Quantitative Methods for Policy Analysis

Course Notes

Nairobi, Kenya

April 28 – May 2, 2014

International Food Policy Research Institute

Course Instructors:

Richard Howitt and Siwa Msangi

Notes Prepared By:

Richard Howitt, Siwa Msangi, and Duncan MacEwan

Table of Contents

Quantitative Methods for Policy Analysis i
Introduction to the Course 1
1 The Development of Computational Economics 1
2 How do we use Economic Models? 2
3 Types of Agricultural Economic Models 3
3.1 A brief taxonomy of model types 3
3.2 Model Specification and Parameter Estimation 5
3.2.1 Econometric Approaches to Specifying Model Parameters 5
3.2.1.1 Positive Degrees of Freedom – the typical case 5
3.2.1.2 Handling Negative Degrees of Freedom – the 'ill-posed' case 6
3.2.2 Model calibration methods 6
3.2.2.1 Application to Constrained Structural Optimization Models 7
3.2.2.2 Application to Macro-level Multi-market Equilibrium Models 7
Day 1: Introduction to GAMS and Linear Programming 9
1 Objective 9
2 Modeling with GAMS 9
2.1 Basic files in GAMS 9
2.2 Basic GAMS Syntax 10
2.3 Basic GAMS Output 12
2.4 Other Useful Code 15
3 Specifying Linear Models 15
3.1 Linear Programming (Primal) 16
3.1.1 Example: Yolo County Farm LP Primal Model – Irrigated Agriculture 16
3.1.1 Example: Machakos Farm LP Primal Model – Non-Irrigated Agriculture 19
3.1.1.1 Investigating the Primal LP Optimal Solution in GAMS 21
3.2 Linear Programming (Dual) 23
3.2.1 The Economic Meaning of the Dual 24
3.2.2 Dual Objective function 24
3.2.3 Dual Constraints 24
4 Positive Mathematical Programming 25
4.1 Behavioral Calibration Theory 25
4.2 A Cost Based Approach to PMP Calibration 26

4.2.1 An Initial Primer on Cost Based PMP Calibration – Single Crop 26
4.2.2 PMP Calibration – Multiple Crops 29
5 Machakos Quadratic PMP Model 33
6 Further Reading 34
Day 2: PMP, Livestock, and CES Production Functions 35
1 PMP Calibration Checks 35
2 Livestock PMP Machakos Model 36
3 CES Production Function 38
3.1 CES Parameter Calibration 39
3.2 CES and Numerical Scaling 41
4 Machakos CES PMP Model 42
5 Calibrating Demands with Limited Data 43
6 Machakos CES PMP Model – Endogenous Prices 43
7 Further Reading 44
Day 3: Dynamic Programming 45
1 Introduction to Dynamic Programming 45
1.1 Simple Analytical Case: Cake-Eating 46
1.1.1 Closed-form analytical example of the Cake-Eating Problem 47
1.1.2 Solution by backward recursion 48
1.1.3 Solution properties of the cake-eating problem 51
1.1.4 Exercises for the Cake Eating Problem 52
2 Value Function Iteration 53
3 Function Approximation 53
3.1 Chebychev Nodes 54
3.2 Chebychev Polynomials 55
4 “Collocation” or Regression 56
4.1 “Chubby”-chev – Eating Cake 58
4.1.1 Polynomial Approximation Exercises 60
5 Application to Senegal Livestock 61
5.1 DP Solution 62
5.1.1 Value Function Iteration and Chebychev Collocation 63
5.2 DP Simulation 63

6 Further Reading 63
Day 4: Stochastic Dynamic Programming 64
1 Introduction to Stochastic Dynamic Programming 64
1.1 Stochastic Cake Eating 64
2 Multi-State Models 66
2.1 Function Approximation 66
3 Agro-Forestry Application 67
3.1 Input Data and State Space 67
3.2 Simulation 68
4 Herd Dynamics Application 68
4.1 Input Data 69
4.2 Simulation 69
5 Application to Groundwater Dynamic Optimization 70
5.1 Input Data 70
5.2 Stochastic State Variables 71
5.3 SDP Solution 71
5.4 SDP Simulation 72
Day 5: Multi-Market Models 74
1 Introduction 74
2 Modeling Inter-regional Trade 74
2.1.1 Primal Trade Model 74
2.1.2 Dual Trade Model 75
2.2 Example: California Water Trading 76
2.2.1 California Water Trading Exercises 76
3 Multi-Market Models 76
3.1 Multi-Market Model Overview 77
3.2 Data Requirements 79
3.3 Example Multi-Market Model: Multi-Good/Region 80
3.3.1 Making the flow of trade endogenous to the model. 81
4 Further Reading 83

Introduction to the Course

As economists, modelers, and other public policy professionals, we are interested in explaining observed actions and using this information to make statistical statements about future states of the world. For example, we may want to predict the adoption of new production technology by farmers and estimate the benefits and costs to the regional economy. Or we may want to predict industry response to new regulations or policy. In order to provide useful insights and effectively guide public policy we require a consistent economic modeling framework.

Historically economists have relied on econometric (or statistical) methods to estimate parameters from observed data. In this approach we observe a rich cross-section or time-series dataset, specify an economic model which implicitly defines the underlying behavior (say, simple linear regression), and estimate key parameters of interest (such as supply and demand elasticities). Econometric analysis typically requires a large dataset and we will often specify a reduced-form model. What do we do when data are limited? What do we do when we want to predict response to policies that simultaneously affect multiple resources, production activities, prices, and markets? Computational methods such as linear programming, calibrated optimization, and dynamic programming allow us to calibrate parameters using limited data and specify a framework that is consistent with economic theory that we can then use to simulate the interaction of complex resource policies.

We want to note that the implied "gap" between econometric and computational methods has been closing rapidly in recent years. With the increasing availability of detailed primal production data, such as satellite-based land and water use estimates, there has been a corresponding trend towards integration of econometrics and calibration. For example, we can use field-level production data to econometrically estimate the physical relationship between crop yield and production inputs and we can use this information directly in the calibrated model. The interested reader should review Merel and Howitt (2014) at the conclusion of the course.

1 The Development of Computational Economics

We begin the course notes with a little background on the development of computational methods and our view for the path forward.

In the past it has usually been the case that econometric models are positive (make statements about what *is*) and programming models are normative (make statements about what *should be*) as they have an explicitly optimized objective function. This has led to an unfortunate methodological division among empirical modelers on the supply side of agriculture and development economics into practitioners of econometric and programming approaches. The difference in approach has been divided along the lines of normative and positive models in the

past. With the development of calibration techniques for optimization models, programming approaches can now incorporate positive data-based parameters and thus build a continuous connection from the pure data-based econometrically estimated models through to the linearly constrained programming models. From one viewpoint, econometric models are data intensive while calibrated nonlinear optimization models are more computationally intensive.

In recent years the sub-discipline of Computational Economics has emerged, principally in the empirical application of macro-economic models. As we would expect, shifts in both the supply and demand have stimulated the emergence of this new economic field. The shift in the supply function of computation ability and cost has largely been driven by "Moore's Law" which states that "the number of transistors on a given chip doubles every eighteen months without any increase in cost." This remarkable trend, which was first proposed by Gordon Moore, a cofounder of Intel, is predicted to continue. Clearly we are in the middle of a dramatic reduction in the cost of computation. Along with the changes in hardware supply, there have been similar changes stimulated in the supply of software for computational economics.

The demand for computational economics is also shifting out due to the increasing complexity and speed required for applied economic analysis. In addition, several of the newer methodologies in stochastic dynamics and game theory are not suited to the analytical process of deriving testable hypotheses often used in conventional estimation methods. As we will see in this course, computational methods are also better suited to handling the complex interaction between economic output and resource availability.

Of more concern to those in this course is the fact that growth areas for applied economic analysis are in agriculture, environmental, resource and development economics. All of these fields are characterized by the absence of large reliable data series and the need for disaggregate analysis. It is not that econometric approaches are unsuited to supply side analysis in these areas, it's just that the data needed are not usually available.

These two shifts bode well for the growth in optimization models in the future. While there are many books on optimization modeling using linear and quadratic structural approaches, for example Paris (1991) or Hazell & Norton (1986). There is no published text on calibrating micro-economic models; the paper provided with your course materials by Howitt et al (2012) provides an excellent overview. This short course and the accompanying notes serve as an introductory text for calibrating optimization models.

2 How do we use Economic Models?

Given an economic phenomenon there are three tasks that we may want to perform with economic models, (i) explain observed actions, (ii) predict or project agent behavior and other economic phenomena that may lie outside of what we observe (under various scenarios), and (iii) influence policy-making by using modeling outputs to illustrate how alternative rules and regulations could result in better economic (and even environmental) outcomes.

We may wish to explain the observed actions. This is usually performed by structural analysis using positive econometric models. Given a structure in the form of a specific set of equations, the parameters that most closely explain the observed behavior are accepted as the most likely explanation.

A second practical use for economic models is in forecasting or predicting economic phenomena arising from the projected behavior of economic decision-makers and agents (acting either individually or in concert with others). As the Druids found out, forecasting significant socio-economic events is a source of power, and can help in the process of planning and resource prioritization. Forecasting models are the ultimate outcome of the positivistic viewpoint where the structure is unimportant in itself and the accuracy of the out of sample forecast is the key determinant of model value. Econometric time series models are the best examples of pure forecasting models, although the ability to produce accurate out-of-sample forecasts should be used to assess the information value of all types of models.

A third use of economic models is to control or influence certain policy-level decisions and rules, so that they can lead to better economic outcomes for all parties involved. This process is generally referred to as policy evaluation since public economic policies are justified on the basis of improving the values of some pre-defined set of economic criteria. Both structural econometric and optimization models are used for policy evaluation, however – due to the relative dearth of good-quality socio-economic data compared to the relative abundance of structural bio-physical data, policy models of agricultural production and resource use are often specified as optimization models.

3 Types of Agricultural Economic Models

There are many types of agricultural economic models that are employed in empirical policy analysis. Given such a large variety, the analyst has to make an informed choice about which one to choose, given that each has a different set of requirements in terms of data needed or computational burden. In this section, we try to lay out some of the key differences more clearly.

3.1 A brief taxonomy of model types

To simplify our discussion on the various types of model that can be used in the analysis of agricultural and resource economic policy problems, we have summarized them into an approximate taxonomy, in the table below, that will help you to understand the range of models that we will discuss within this course.

	Static	**Dynamic**
Micro-level	(household or commercial) Farm production models	(renewable or non-renewable) Resource management models
Macro-level	Multi-market (partial- or general-) equilibrium models	Macro-economic growth models

Depending upon the nature of the problem considered, we might want to just consider the behavior of economic agents in one period – in which case a *static* model is adequate for our analysis – or else we can employ a *dynamic* model if we want to consider the consequences that the behavior of agents in one period has on outcomes in subsequent periods. Many policy analyses only want to look at the effect of a single intervention and compare the 'with' to the 'without' cases, and its consequences on welfare or some specific economic decision (or set of decisions) in a 'snapshot'. A static analysis can provide useful information that can point to the relative efficacy of one policy or intervention, in relation to another. In the case of natural resource problems (typically) or other problems in which there is a "stock" of resources that can be accumulated (or depleted) over time – the dynamic effects of how agent behavior impacts the stock of that resource over time matters. The quantitative methods for this kind of analysis are typically more computationally intense, since they must add the dimension of time to the set of factors (space, agent types, etc) already under consideration. We will consider alternative ways of handling these problems, analytically.

The analyst also faces the choice of whether to consider the micro-level behavior of decision-makers at the household (or 'enterprise') level, versus the macro-level outcomes resulting from multiple agents interacting through price-mediated market forces. This choice depends, of course, on the types of outcomes that one wishes to observe (farm or land-scape level versus sector or economy-wide level), and the level at which the relevant decision-maker (whom the analyst hopes to study or influence) acts upon the economic or physical environment. The analyst typically gets greater "resolution" and detail, when considering the behavior of individual agents at micro-scale, given that the nature of bio-physical interactions between the agricultural production (or resource extraction) activities and the environment are much more explicit, compared to the 'birds-eye' view provided by more macro-scale analyses. When looking at a more aggregate, macro-scale level, the analyst often has to reduce the decision-making process and underlying behavioral characteristics and 'rules' of the agents into reduced-form equations that can be solved simultaneously with the equations governing the reaction or behavior of other agents in the economy. This makes capturing the details of bio-physical interactions between socio-economic behavior and the environment more difficult, and embodies a typical tradeoff that analysts face when deciding upon what kind of model should be employed to answer a particular research question.

3.2 Model Specification and Parameter Estimation

Once the type of model has been chosen, the analyst faces another challenge in finding the values of the behavioral parameters that can operationalize the models and allow them to (plausibly) capture the decision, actions and re-actions of economic agents to external forces of change in the physical or socio-economic environment. Given that the analyst wishes to represent the behavior of agents with some fidelity, and use it within the model-based analytical framework to better project possible behavior under alternative policies or socio-economic forces, in an *ex ante* fashion – he or she will typically make use of data on past behavior in order to establish the relationship between the drivers of change and the outcomes of economic agent behavior.

We can classify the approaches to doing this into 2 broad categories – namely: (1) econometric estimation, on the one hand, where the range of available data can be used to estimate behavioral relationships for a single representative agent (or a range of agent types) that can help validate the model over a particular period or range of observations ; or (2) calibration – where we want to "anchor" the model to reproduce observed behavior at one point in time, and use it as a basis for dynamic (or static) simulations without as much emphasis on the statistical robustness of the parameter estimates in the behavioral equations. We discuss these approaches, briefly, in the following sub-sections.

3.2.1 Econometric Approaches to Specifying Model Parameters

We review econometric approaches to specifying model parameters. Econometric methods typically require a richer dataset, which may not be readily (or consistently) available for many applications.

3.2.1.1 Positive Degrees of Freedom – the typical case

Estimating the structural parameters of an economic model through econometric estimation techniques has been the standard approach to specifying agricultural economic models for the past twenty years. Econometric estimates of model parameters – that can make use of ample observations of agent behavior (either in cross-section or over time) -- offer the analyst the opportunity to test and employ more flexible and theoretically consistent specifications of the technology (or of preferences) than is typically possible with programming models – where plausible parameters are usually 'chosen' by the analyst. In addition, econometric methods are able to test the relevance of the constraints and parameters in the model specification given an adequately-sized data set, and allow the analyst to derive measures of statistical significance, goodness-of-fit or robustness of the overall estimation. The initial econometric research on production models was performed on aggregated data for multi-output / multi-input systems, or single commodities for more disaggregated specifications. However, despite several methodological developments, econometric methods are rarely used for disaggregated empirical microeconomic policy models of agricultural production. This is usually because time series data

is not generally available at a high level of disaggregation and the implicit assumptions needed for cross-section analysis are not usually acceptable to policy makers with regional constituencies. In short, flexible form econometric models have not fulfilled their empirical promise mostly due to data problems that do not appear to be improving.

3.2.1.2 Handling Negative Degrees of Freedom – the 'ill-posed' case

If we wish to use sample data in a way that doesn't impose a single agent-type – which is implicitly the case, when we try and estimate the parameters for a 'representative' farm or household from a dataset that captures the actions of a diversity of agents in the real economy – then we quickly run into a situation where we might need to solve for a greater number of parameters than what we have data points for. In other words, we have *negative* degrees of freedom. There is a class of econometric methods that can handle this case – which make use of information theory, and which allow the analyst to incorporate his or her best guess or "prior" into the estimation process, similar to how it is done within classical Bayesian statistics or econometric methods.

The application of this class of estimation procedures to the estimation of model parameters and behavioral equations is still relatively new in the literature and beyond the scope of this course. Briefly, this approach enables consistent reconstruction of detailed flexible form models of cost or production functions on a disaggregated basis. This requires that the model contain more parameters than there are observations – hence the term "Ill-Posed" problems. An application to micro production in agriculture can be found in Paris and Howitt (1998) and Howitt and Msangi (2014), included with your supplemental course reading material.

3.2.2 Model calibration methods

Much of this course is focused around a method of calibrating structural programming and equilibrium models in a way that makes use of data that we observe (i.e. in a "positive" manner) – but which do not need to rely on large quantities of observations or extensive socio-economic data sets. In such cases, we only need to have as many observations as the number of parameters that we're trying to solve for – in which case, we have *zero* degrees of freedom. The approach that we will illustrate for the micro-level farm production models that we will examine in this course, uses the observed allocations of crops and livestock to derive nonlinear cost functions that calibrate the model without adding unrealistic constraints. The approach is called Positive Mathematical Programming (PMP) (Howitt 1995). In part of this course we will learn to specify, solve and interpret several micro-level, Positive and Normative Programming models used in Agricultural and Environmental Economics. We will also consider calibration techniques applied to macro-level models in which there may not be an explicit formulation of the underlying, hypothetical optimization problem that governs the behavior of agents – but which have reduced-form behavioral equations that must be solved simultaneously for various agents and sectors.

3.2.2.1 Application to Constrained Structural Optimization Models

The archetypical characterization of an economic problem is where an agent tries to optimize with respect to some socio-economic objective in the presence of (economic or physical) constraints. When considering problems at a micro-level, where the household or economic enterprise is limited in some way by critical factors (such as its own labor or the availability of land, capital or credit), an optimization-based analytical framework can be very useful in understanding the tradeoffs the decision maker faces, and how best they can be mitigated. Optimization models have a long history of use in agricultural economic production analysis. There is a natural progression from the partial budget farm management analysis that comprised much of the early work in agricultural production to linear programming models based on activity analysis and linear production technology. Often linear specifications of agricultural production are sufficiently close to the actual technology to be an accurate representation. In other cases the linear decision rule implied by many Linear Programming (LP) models is optimal due to Leontief or Von Liebig technology in agriculture.

Despite the emphasis of methodological development for econometrically-derived structural economic models, programming models are still the dominant method for micro-analysis of agricultural production and resource use. Their applications are widespread due to their ability to reproduce detailed constrained output decisions and their minimal data requirements. As noted above, econometric model applications on a microeconomic basis are hobbled by extensive data requirements.

LP models are also limited largely to normative applications as attempts to calibrate them (and therefore apply to positive analysis) to actual behavior by adding constraints or risk terms have not been very successful. Much of the critique that has been made of programming-based approaches, in the past, have come about as a results of the relatively poor performance of pure LP-based models to replicate observed behavior. The calibration methods that we will discuss in this course, address many of these problems, and have helped bring programming-based structural models back into the mainstream of applied agricultural policy analysis.

3.2.2.2 Application to Macro-level Multi-market Equilibrium Models

If we move beyond micro-level applications at farm- or landscape-level, and consider the behavior of multiple agents acting simultaneously and reacting to each other through price-mediated market interactions – we are in the realm of multi-market equilibrium models, which are either partial- or general-equilibrium in nature. Much analysis of agricultural and natural resource policy problems has been done within a partial-equilibrium framework, in which the analysts attention is restricted to the particular sector (or sectors) of interest – without worrying about how the markets for all the relevant factors of production interact with the production, and how they feedback (through income relationships) to the consumption side of the market. Applied general-equilibrium (typically called “computable general equilibrium” or CGE) models typically follow along those lines, and take a much broader, economy wide perspective, in order to account for the circularity of income flows within the economy, and how the revenues of

producers translate into incomes for households or tax revenue for the government. A partial-equilibrium multi-market or sectoral model can be calibrated by calculating the intercepts of the supply and demand functions, such that the model reproduces the base period of the simulation when it is solved for a given set of prices and observed supply, demand and trade values. Within a general-equilibrium framework, this base year solution must also reproduce the flows of income from one sector of the economy to the other – such that wage payments, tax revenues, export/import values and the value of production and consumption across all sectors (and agents) within the economy are accounted for. The Social Accounting Matrix (SAM) is at the heart of a CGE model and is constructed from a standardized system of national accounts and input-output that has been developed over time to conduct economy wide macro-economic assessments. The values of production and consumption that are embodied within the same are used as the basis for computing the intercepts of supply and demand functions, that allow CGE models to calibrate to their base period, much in the same way as is done for partial-equilibrium multi-market or sector models. We will use the example of a simple partial-equilibrium, multi-market model to illustrate the calibration process, later in the course. Criteria for Model Selection

Selection of the best model for the research task at hand is an art rather than a science. The model builder is constantly balancing the requirements of realism that complicate the model specification and solution against the practicality of the model in terms of its data and computational requirements. This trade-off is similar to the selection of the optimal photographic models for a website where the publisher has to make the subjective tradeoff between the beauty of the model and the degree of realism that the model will portray. The optimal customer response to the website will come from models who are eye-catching but with whom the customers can identify. In economic policy models, they must be simple enough so that the decision maker can identify with the model concept, but at the same time be tractable and able to reproduce the base year data.

There is no ideal model, just some that are more manageable and useful than others. Hopefully, this course will give you a starting-point for the theoretical and empirical tools to make informed decisions on the best model specification for particular data and research situations.

The process of econometric model building has three well-defined stages: Specification, Estimation, and Simulation. Programming model building methods have not formally separated these stages. The equivalent stages are Specification, Calibration and Policy Optimization. However the important process of calibrating the models is usually buried in the model specification stage. Howitt et al (2012) review the optimization model building stages. This course will explicitly address these different stages of optimization model building and differs from the usual treatment by having a strong emphasis on model calibration.

Day 1: Introduction to GAMS and Linear Programming

1 Objective

At the end the day you will be able to:

a. Use both quantity and price data on inputs and outputs to define a consistent model of rural water use.
b. Calibrate and run regional or farm models from minimal data sets.
c. Use the models to calculate the regional water demands and the elasticity of demand.
d. Estimate the value of rural water demands for water policy.

2 Modeling with GAMS

Before we dive into the programming models we want to stop and review the basics of the GAMS programming language. GAMS stands for the General Algebraic Modeling System and it is a matrix-based computer software package specifically designed to solve systems of equations. We have provided an introduction to GAMS programming with your course materials, and suggest you familiarize yourself with the introductory sections. We review some of the key concepts here.

We will use the GAMS-IDE (Integrated Development Environment) in this course. GAMS-IDE is essentially a user-friendly graphical interface. We are still able to see our programming code, but all of the solvers run in the background and all output is nicely (once you get used to it!) organized.

2.1 Basic files in GAMS

- The project file (xxxx.gpr)
    - We will create this file the first time we start a new GAMS program. This file tells GAMS where to look for the program file and output file.
- The program file (xxxx.gms)
    - This is where we will write the code for all of our problems in the course. We will also use this file to store all of our input data. You will use this file to "run" the program. After the program runs it will automatically create an output file.
- The output file (xxxx.lst)

- This file contains all of the important output parameters and information. GAMS-IDE will output all of the parameters, variables, equations, and solutions that you will specify. It can be a large amount of information but fortunately GAMS-IDE will also generate a "table-of-contents" that you can use to quickly navigate to important output.

Getting Started Part 1:

1. Save the Intro program file in a folder of your choosing
2. Open GAMS-IDE
3. Go to file – project – new project
4. Navigate to the folder where you placed the program file and create the project file
5. Go to file – open – select your program file
6. Should now have your Intro program file (code) open

2.2 Basic GAMS Syntax

The program file is the heart of analysis using GAMS and we will become very familiar with this file before the end of the course. GAMS looks for certain types of information in the program file. It is helpful to understand what types of information we will provide and the logical order.

Hint: GAMS is not case sensitive.

Hint: Use "" to comment out lines of code*

Hint: Notice placement of ";" – they are important!

First we consider basic data management in GAMS. The GAMS program file will contain all of the data for the program.

- Sets
  - We define all sets at the beginning of the program file. Sets are used to tell GAMS what to expect in the way of elements in the data (parameters, tables, and scalars) , variables (variable, positive variable) , and model structure (equations, model, solve).
- Data: organized by three methods
  - Tables
    - Must be defined over sets and organized as rows and columns to read in input data.
    - Syntax: TABLE *name* description {list rows/columns as a matrix} ;
  - Parameters

        - Similar to a table, we will use this to organize input and output data.
        - Syntax: PARAMETER *name* description /values/ ;
    - Scalars
        - Dimensionless
        - Syntax: SCALAR *name* description /value/ ;

Getting Started Part 2:

7. Start at the top of the Intro program file
8. Locate the SET definitions, follow the comments in the code and define a new set
9. Locate the input data TABLE and review the syntax
10. Locate the input PARAMETER, follow the comments in the code to define a second parameter
11. Locate the SCALAR input data

Now, let's move to the model definition section of GAMS. We will define a model as a series of variables and equations. The model will use the input data you previously specified and will be a simple (and boring) farm profit maximization problem.

Hint: Colors in GAMS help you determine when you have made a programming error

- Variables: include factors that are unknown and endogenous to the program.
    - Variable – standard problem variable, endogenous.
    - Positive variable – restricted to positive values
    - Other types – not considered in this course
- Characteristics of variables: you may see these after a variable and they indicate a specific operation.
    - Upper bound .UP
    - Lower bound .LO
    - Level .L
    - Dual value .M
    - Fixed value .FX
- Equations
    - We use these to define the constraints and objective function
- Model

- We use the MODEL statement to give our model a name and tell GAMS which equations we want to include (typically all). We can have multiple models defined within the same program file.
  - Syntax: MODEL *model_name* /equations/ ;

- Solve
  - We use the SOLVE statement after we define the model. This tells GAMS to solve the program!
    - Syntax: SOLVE *model_name* using *solver_name* MAXIMIZING *variable_name* ;

Getting Started Part 3:

12. Review the variable and define a second variable
13. Review the equation definitions
14. Define the model following the comments in the code
15. Review the solve statement

2.3 Basic GAMS Output

We have now defined a simple farm profit maximization program in GAMS. The last thing we need to do is define the output from the model. We can define output parameters where we summarize or calculate key output. You will see examples throughout the course. Here we will simply use the display statement to output the optimized variable to the output file.

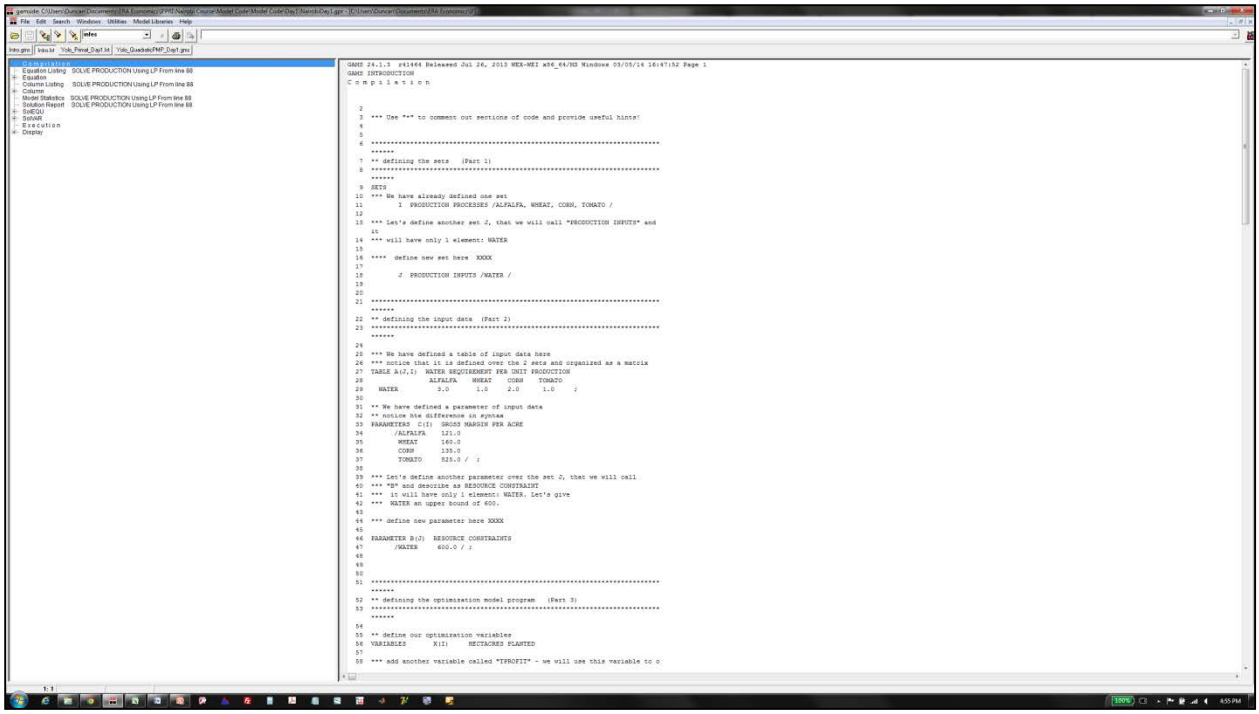
Getting Started Part 4:

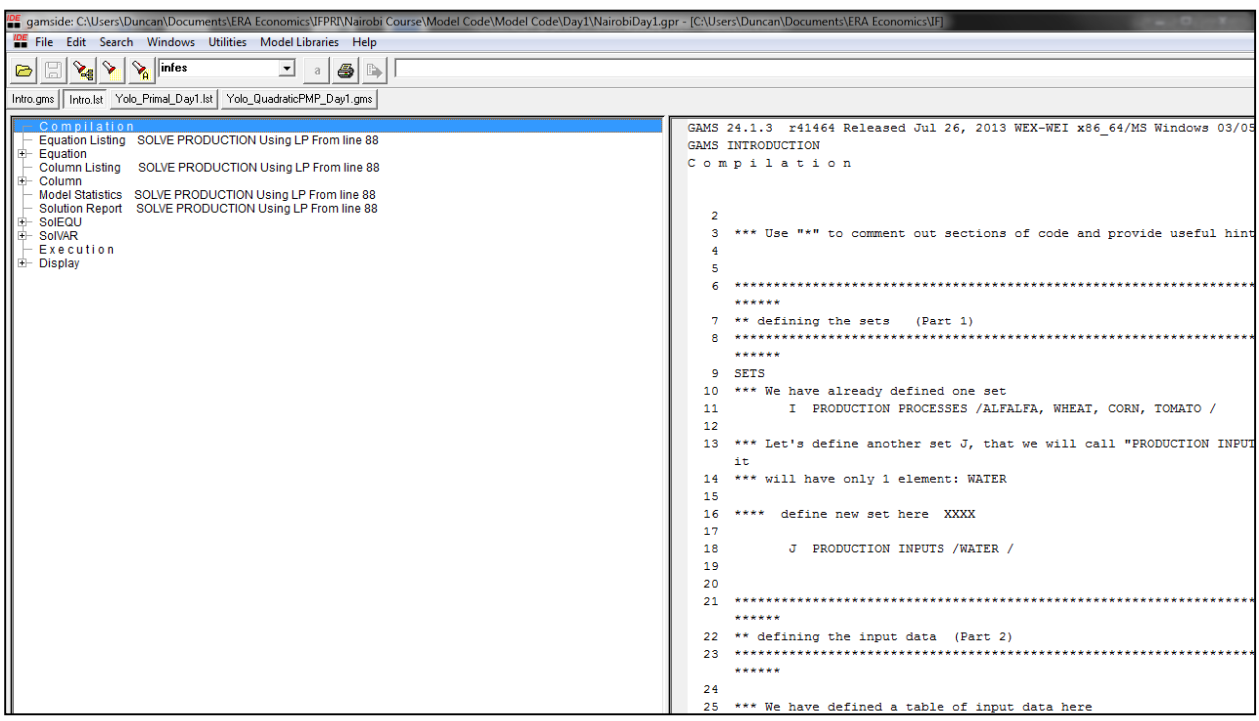
16. Review the DISPLAY statement
17. Run the program!

- Display
  - Statement that tells GAMS to display information.
    - Can include data that you have defined, output variables, or information about constraints or other model equations.

After we run the program GAMS generates a "process window" which briefly summarizes the status of the solve attempt. Errors are shown in red, if any. If there are no errors, close the window.

GAMS will automatically generate the output file (.lst). The output file contains a large amount of information and also creates a table of contents on the left. We will use the table of contents to navigate the file.





The output file includes:

- A copy of the entire program file with line numbers
- Equation and column listing
- Model statistics

  - Solution report (technical solver information)
- Solution Equations
  - Solved equation summary information
- Solution Variables
  - Solved variables summary information

The output file contains useful information for each equation and variable. We will discuss what these mean during the course. For now we simply need to know where they are located.

- Lower
  - Lower bound
- Level
  - Optimized level
- Upper
  - Upper bound
- Marginal
  - Dual value

We can see the RESOURCE and variable X output information from the GAMS output file in the screenshot below. We will learn how to interpret the various components of output information in the context of a farm profit maximization model later today.

```
IBM ILOG CPLEX   24.1.3 r41464 Released Jul 26, 2013 WEI x86_64/MS Windows
Cplex 12.5.1.0

LP status(1): optimal
Cplex Time: 0.00sec (det. 0.00 ticks)
Optimal solution found.
Objective :      495000.000000


---- EQU RESOURCE  AVAILABILITY CONSTRAINTS OF RESOURCES

         LOWER     LEVEL     UPPER    MARGINAL

WATER     -INF    600.000   600.000   825.000

                      LOWER     LEVEL     UPPER    MARGINAL

---- EQU PROFIT          .         .         .      -1.000

  PROFIT  PROFIT DEFINITION

---- VAR X  HECTACRES PLANTED

           LOWER     LEVEL     UPPER    MARGINAL

ALFALFA      .         .        +INF  -2354.000
WHEAT        .         .        +INF   -665.000
CORN         .         .        +INF  -1515.000
TOMATO       .      600.000     +INF       .

                      LOWER     LEVEL     UPPER    MARGINAL

---- VAR TPROFIT        -INF  4.9500E+5     +INF
```

2.4 Other Useful Code

We can use "$ontext" and "$offtext" to comment out sections of code within the program file. Check the program files provided with your course materials for some examples. We use the following syntax:

$ONTEXT

Text text text

Text text etx

$OFFTEXT

We will use loops to speed up policy simulations. We can define loops within GAMS to sequentially solve the model over a predefined set. We use the following syntax:

LOOP (*SET*,

--- loop items

);

If we want to define sets "on-the-fly" we can use "*" in the parameter definition. This is particularly useful when we want to summarize output from a model solve statement within a loop. For example,

PARAMETER RESULTS(*,I) Model Output ;

Then we can define the first set on the fly to describe relevant output parameters

RESULTS("NEWSET",I) = X.L(I);

We have covered the basics of the GAMS program syntax and output. The best way to understand the language is to write code, make mistakes, fix mistakes, and repeat the process.

How do we deal with errors? This is a course unto itself and we will refer the interested reader to the GAMS notes provided with your course materials. That or the helpers walking around the classroom!

3 Specifying Linear Models

Now that we have a little GAMS syntax primer under our belt we will start to develop the Linear Programming problem. We begin with a little theory and then work through the "Primal" and "Dual" specifications of the problem. We emphasize the primal formulation and only discuss the dual problem in the context of important information we can extract.

3.1 Linear Programming (Primal)

Define crop hectares (x_i) by the vector $\mathbf{x}$. As we will see later in the course, we can also think of this as livestock numbers. Each hectare yields (y_i) which can be sold at price (v_i), defined by the vectors $\mathbf{y}$ and $\mathbf{v}$, respectively.

We will assume Leontief production technology. We introduce a matrix of technical coefficients, a_{ij}, where these tell us the required amount of input (resource) j per hectacre. We will define our matrix of technical coefficients as $\mathbf{A}$. Each resource has a cost cs_j, defined by the vector $\mathbf{cs}$.

To simplify the problem and notation we will sometimes define the gross margin per hectare as

$$(1) \qquad \mathbf{c} = \mathbf{y}'\mathbf{v} - \mathbf{A}\,\mathbf{cs},$$

where gross margin tells us the returns per hectare net of production costs. This is particularly useful for problems with Leontief technology. We can write the LP as,

$$(2) \qquad \text{Max } \mathbf{c}'\mathbf{x}$$

$$(3) \qquad \text{subject to } \mathbf{A}\mathbf{x} \le \mathbf{b}$$

We can state the LP problem as follows: "we want to find the non-negative values of a vector $\mathbf{x}$ which satisfies the constraints and maximizes the objective function." The objective function is the sum of gross margin per hectare across all hectares. The constraints define the technical resource requirements and availability. The notation generalizes to any number of inputs, outputs, and constraints.

We can specify the LP as a profit maximization problem (as shown in this section) or, equivalently, as a cost minimization problem. Both will yield the same optimal solution but require different data to estimate the model. These are known as the primal and dual problems, respectively, and one can be easily derived from the other. We will illustrate the primal problem and then the dual problem in the subsequent sections.

3.1.1 Example: Yolo County Farm LP Primal Model – Irrigated Agriculture

As a working example in this course we will use a farm problem based on agriculture in the Machakos study area. In the course notes we will first present an example of irrigated agriculture using a model of agriculture in Yolo County, California. We hope that this provides you with a basis for irrigated and non-irrigated farm production models, once you get past the funny units of measurement.

Yolo County is an area in Northern California with a diverse mix of irrigated agriculture. The ideas presented in this example readily extend to situations in developing countries and non-irrigated agriculture. We will apply these concepts to the Machakos model.

The example farm has the possibility of producing four different crops, alfalfa, wheat, corn and tomatoes. Yields are fixed so we can measure output by the number of acres of land allocated to each crop. The variable costs have been subtracted for simplicity and we focus on gross margin per acre. Constraints on production are all inequalities and represent the maximum amounts of land, water, and labor available, and a contract marketing constraint on the maximum quantity of tomatoes that the farmer can sell in any year.

The resulting linear program can be written as follows:

Maximize the scalar product of gross margin per acre (remember that **c** is (n by 1) and **x** is also (n by 1)):

(4) Max $\mathbf{c}'\mathbf{x}$

(5) subject to $\mathbf{Ax} \leq \mathbf{b}$

The vector of production or activity levels, measured in acres of land, is:

(6) $\mathbf{x}' = [x_1 \ x_2 \ x_3 \ x_4]$

(7) $x_1 =$ Alfalfa, $x_2 =$ Wheat, $x_3 =$ Corn, $x_4 =$ Tomato

We have already subtracted out production costs, so we can measure activity profitability in terms of net returns (dollars per acre). The vector of activity level gross margin is:

(8) $\mathbf{c}' = [c_1 \ c_2 \ c_3 \ c_4]$

(9) $c_1 = \$121/\text{acre},\ c_2 = \$160/\text{acre},\ c_3 = \$135/\text{acre},\ c_4 = \$825/\text{acre},$

You should verify that when we multiply out we can write the objective function as:

(10) $$\left[121x_1 + 160x_2 + 135x_3 + 825x_4\right]$$

Next, we need to define the maximum amount of inputs available. We specify these in physical units:

(11) $$\mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \equiv \begin{bmatrix} \text{Land (acres)} \\ \text{Water (acre-feet)} \\ \text{Labor (hours)} \\ \text{Contract (tons)} \end{bmatrix}$$

(12) $b_1 = 600$ acres, $b_2 = 1{,}800$ acre-feet, $b_3 = 5{,}000$ hours, $b_4 = 6{,}000$ tons,

Finally, we need to define the matrix of technical coefficients. Recall that these coefficients translate inputs (*j*) into output, and are therefore in the same physical units as the input per unit output. For example, the technical coefficients describing the use of water (acre-feet) in the production of output (in this example, acres of the 4 crops) would be in acre-feet per acre. We will return to this point, first we define:

$$(13) \qquad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4.5 & 2.5 & 3.5 & 3.25 \\ 6 & 4.2 & 5.6 & 14 \\ 0 & 0 & 0 & 33.25 \end{bmatrix}$$

Next we define the full LP constraint and then interpret the coefficients. We can write the constraint (where we have included units for clarity) as:

$$(14) \qquad \mathbf{Ax} \le \mathbf{b} \iff \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4.5 & 2.5 & 3.5 & 3.25 \\ 6 & 4.2 & 5.6 & 14 \\ 0 & 0 & 0 & 33.25 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \le \begin{bmatrix} 600 \text{ acres} \\ 1{,}800 \text{ acre-feet} \\ 5{,}000 \text{ hours} \\ 6{,}000 \text{ tons} \end{bmatrix}$$

Using matrix multiplication we can see the constraint set of our problem. Consider the first row of $\mathbf{A}$, which defines our land constraint. We can multiply out and get:

$$(15) \qquad 1x_1 + 1x_2 + 1x_3 + 1x_4 \le 600 \text{ acres}$$

The constraint tells us that each acre of crop requires an acre of land. A boring, yet intuitive constraint! We can see that there are a total of 600 acres to be allocated on the farm. Let's turn to the second (and more interesting) constraint:

$$(16) \qquad 4.5x_1 + 2.5x_2 + 3.5x_3 + 3.25x_4 \le 1{,}800 \text{ acre-feet}$$

The constraint tells us that we have a total of 1,800 acre-feet of water to irrigate crops. Each crop has a different water requirement, as reflected in our matrix of technical coefficients. We see that crop x_1, which happens to be alfalfa, requires 4.5 acre-feet per acre, whereas crop x_3, corn, requires 3.5 acre-feet per acre. Summing across all crops we get the total water use on the farm, which must be less than the total available water (1,800 acre-feet). As an exercise you should multiply out and interpret constraints 3 and 4.

We can solve the model by hand, or use GAMS, and find the optimal solution is:

$$(17) \qquad \mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \end{bmatrix} = \begin{bmatrix} 0 \\ 419.54 \\ 0 \\ 180.45 \end{bmatrix}$$

In other words, the optimal solution is to grow 419.54 acres of wheat and 180.45 acres of tomatoes and no other crops. Is this consistent with our expectations? We can verify that the solution is consistent with profit maximizing behavior.

We expect tomatoes to come into the profit maximizing solution since they have a high profit margin per unit of land. In fact, we should grow as many acres of tomatoes as possible since they are the most profitable crop. Let's explore this a little further.

Looking at constraint 4 we see that on one acre we can grow 33.25 tons of tomatoes but can only sell 6,000 tons to the processor. What is the maximum number of tomato acres we can plant? We can compute the maximum number of acres we can plant by dividing the contract constraint by yield per acre:

$$(18) \qquad \frac{6,000 \text{ tons}}{33.25 \text{ tons/acre}} = 180.45 \text{ acres}$$

Notice this is exactly what the optimal solution shows: we should allocate 180.45 acres to tomatoes. Intuitively, land will next be allocated to the next most profitable crop until we run out of either land or other resources. In this example, we allocate all remaining land (419.54 acres) to the next most profitable crop (wheat) because we have excess amounts of the other resources (water and labor).

3.1.1 Example: Machakos Farm LP Primal Model – Non-Irrigated Agriculture

We will now consider the Machakos farm primal LP example. We will follow this example through for much of the course. The GAMS code for this problem is included in the Day 1 files provided with the course material.

Follow along with file: Machakos_Primal_Day1.gms.

The vector of production or activity levels, measured in hectares of land, is:

$$(1) \qquad \mathbf{x}' = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]$$

(2) $x_1 =$ Inter Cropped, $x_2 =$ Maize, $x_3 =$ Beans, $x_4 =$ Tomato, $x_5 =$ Grass

The vector of activity level gross margin (KSh/ha) is:

$$(3) \qquad \mathbf{c}' = [c_1 \ c_2 \ c_3 \ c_4 \ c_5]$$

$$(4) \qquad c_1 = 13,563/\text{ha}, \ c_2 = 8,350, \ c_3 = 31,125, \ c_4 = 37,704, \ c_5 = 24,980$$

Our resource (input) constraints are defined in physical units as:

$$(5) \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \equiv \begin{bmatrix} \text{Land (hectares)} \\ \text{Labor (person days)} \\ \text{Chemicals (kg)} \\ \text{Seed (kg)} \end{bmatrix}$$

$$(6) \qquad b_1 = 2.78 \text{ ha}, \ b_2 = 250.0 \text{ pd}, \ b_3 = 6,000 \text{ kg}, \ b_4 = 6,000 \text{ kg}$$

Finally, we can define our matrix of technical coefficients.

$$(7) \qquad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 40.3 & 159 & 126.5 & 136 & 0 \\ 8.75 & 83.9 & 12.03 & 181.3 & 30 \\ 43 & 44.6 & 50.3 & 22 & 0 \end{bmatrix}$$

Again, we can write-out the constraints and interpret. Consider constraint three which tells us the total amount of chemicals available for application on the farm. We can write

$$(8) \qquad \begin{aligned} & a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 \le b_3 \Leftrightarrow \\ & 8.75x_1 + 83.9x_2 + 12.03x_3 + 181.3x_4 + 30x_5 \le 6,000kg \end{aligned}$$

The constraint tells us the maximum amount of chemicals (in kg) available on the farm. We see that there is a total of 6,000kg of chemicals (fertilizers, pesticides, etc) available to be allocated to the crops. We have aggregated all chemical inputs for simplicity, but the LP framework can easily handle a model with individual inputs. The matrix of technical coefficients tells us the chemical application rates for each of the 5 crops. Since we have specified a Leontief technology these inputs are applied in fixed proportions per hectare of land. For example, crop 2 (maize) requires 83.9 kg/ha of chemical application and crop 3 (beans) requires 12.03 kg/ha. By multiplying by planted hectares and summing across all crops we get the total chemical use on the farm.

As an exercise you should:

- Multiply out constraints 1, 2, and 4
- Interpret and verify units

When we run the Machakos model in GAMS we see that the optimal solution is

$$(9) \qquad \mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \\ x_5^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.838 \\ 0.942 \end{bmatrix}$$

We allocate no land to intercropping, maize, or beans and plant the farm to a mix of tomatoes and Napier grass. Why does this happen? As with the Yolo irrigated production model in the previous section, we see that we allocate land to the highest value crops until resource constraints become binding. By inspecting the gross margin per hectare $(\mathbf{c})$ we see (again) that

$$(10) \qquad c_1 = 13,563/\text{ha},\ c_2 = 8,350,\ c_3 = 31,125,\ c_4 = 37,704,\ c_5 = 24,980$$

The highest value per hectare is crop 4 (tomatoes) with a gross margin of 37,704 KSh/ha. We grow 1.838 ha of tomatoes. We do not plant the entire farm to tomatoes because a resource constraint becomes binding. With 1.838 ha of tomatoes we are using up all the available labor person days (250). We can see this:

$$(11) \qquad 1.838\,\text{ha} \times 136\,\text{pd/ha} \overset{\text{(due to rounding error)}}{\approx} 250\,\text{pd}$$

Therefore once we have planted 1.838 ha of tomatoes we have used up all available labor. The next most profitable crop is beans at 31,125 KSh/ha gross margin. However, we see that one hectare of beans requires 126.5 pd of labor. Since we do not have labor available we must find another crop. Napier grass has a lower gross margin of 24,980 but requires no labor hours. Thus, the optimal solution is to allocate the remaining land to gross (0.942 ha). We see that the total land constraint is also binding and we use all 2.78 ha of land available on the farm.

This raises some interesting questions that we will investigate later in the course. Why does our LP tell us the optimal solution is x* but we observe the farmer growing a different crop mix? Is the farmer wrong in his acreage decisions or are we missing important information? Specifically, we see that:

$$(12) \qquad \mathbf{x}^* = \begin{bmatrix} x_1^* \\ x_2^* \\ x_3^* \\ x_4^* \\ x_5^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.838 \\ 0.942 \end{bmatrix}$$

But we observe base land use $(\mathbf{XBASE})$

$$(13) \qquad \mathbf{XBASE} = \begin{bmatrix} x_1^B \\ x_2^B \\ x_3^B \\ x_4^B \\ x_5^B \end{bmatrix} = \begin{bmatrix} 0.73 \\ 0.71 \\ 0.47 \\ 0.21 \\ 0.66 \end{bmatrix}$$

And we want to reconcile these differences. If we want to use the model for policy analysis (for example, to evaluate the effects of a change in the price of a crop) we require the model to replicate the base solution. We will return to this point when we introduce the concept of PMP.

3.1.1.1 Investigating the Primal LP Optimal Solution in GAMS

We can dig a little deeper into the solution of the Primal LP program and get a feel for the output from the GAMS solver.

We saw that the solution of the LP was to allocate the maximum land to tomatoes because it was the most profitable crop. Then we allocate the remaining land to Napier grass because it has no labor requirement. Go to the solution (.lst) file and select the "RESOURCE" equation under the heading "SolEQU." You should see the output summary table reproduced below.

	LOWER	LEVEL	UPPER	MARGINAL
LAND	-INF	2.780	2.780	24,980.00
LABOR	-INF	250.00	250.00	93.55
CHEM	-INF	361.52	6,000	.
SEED	-INF	40.44	6,000	.

This table summarizes the resource constraints in the LP model:

LOWER: tells us the lower bound on resource availability. Note that there is no lower bound on resource usage in this example (it is implicitly zero).

LEVEL: tells us the optimized level of resource usage, in other words the vector resulting from $\mathbf{Ax}$. For example, we see that total land use across all crops on the farm was 2.78 hectares. We can see that 250 person days of labor were used.

UPPER: tells us the upper bound on resource availability. This should be familiar to you as our vector $\mathbf{b}$.

MARGINAL: tells us the "shadow" or "dual" value for each resource constraint. This is an important concept in economics and in this course in particular. We will return to these values in the subsequent section and frequently throughout the course. The shadow value is the implicit (not observed by the market) value of a resource as reflected in the change in profit from its productive use. In other words, it is the marginal willingness to pay for another unit of that resource. Mechanically, the shadow value is the increase in the objective function if we had one additional unit of the resource. For example, our solution shows that if we had one additional hectare of land, we would be able to make an additional 24,980 KSh profit.

Notice that there is only a shadow value associated with the binding resource constraints. In our example land and labor are the two binding constraints. You can investigate changes in the resource constraints and see the corresponding change in optimal solution.

We are starting to see the "rigid" nature of the LP formulation – we produce the highest value crops up until resource constraints become binding. How realistic and useful is such a framework for policy analysis? What if we observe the farmer growing something other than the "optimal" solution of tomatoes and Napier grass? How do we get our program to reproduce observed behavior? We will start to investigate some of these questions later today and throughout Day 2.

3.2 Linear Programming (Dual)

The problem of minimizing the cost of inputs subject to constraints on a minimum output level is equivalent to the problem of maximizing profit subject to production technology and constraints on the total input available. Thus every optimization problem can be posed in its Primal or Dual form. For every Primal Problem there exists a Dual Problem which has the identical optimal solution. So far in this course we have only dealt with the primal form of the problem since its intuitive explanation is easier.

We will not cover the dual problem in the course lecture and GAMS exercises. We will only discuss the dual program in the context of the dual values on the resource constraints. This section of the notes is provided for completeness and for the reader interested in a little deeper understanding of the dual program.

We define our familiar vector x (n × 1) of primal variables and now define λ as the corresponding (m × 1) vector of dual variables. There is a dual variable associated with each resource constraint. The "dual" or "shadow" value tells us the marginal change in profits from an additional unit of the resource. This value is only positive if the resource constraint is binding. Intuitively, we are only willing to pay for an additional unit of a resource that is in short supply.

The standard form of the twin Primal and Dual problems can be written as follows:

a. Primal

(14) Max $\mathbf{c}'\mathbf{x}$

(15) subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$

(16) $\mathbf{x} \geq \mathbf{0}$

b. Dual

(17) Min $\boldsymbol{\lambda}'\mathbf{b}$

(18) subject to $\mathbf{A}'\boldsymbol{\lambda} \geq \mathbf{c}$

(19) $\boldsymbol{\lambda} \geq \mathbf{0}$

The Objective functions ask the following questions:

a. Primal
   i. What is the maximum value of the firm's output?
b. Dual
   i. What is the minimum acceptable price that I can pay for the firm's assets?

The dual specification of a problem is particularly useful:

a. When the Dual specification is simpler to solve than the Primal specification
b. When you know production costs but not production technology
c. When it's the dual values that interest you (often for Economists)

3.2.1 The Economic Meaning of the Dual

The Dual Variables λ_i are elements in the vector of imputed marginal values of the resources b_i. Equivalent intuitive interpretations are the opportunity costs of not having the last unit of resource, or equivalently, how much you would pay for one more unit of the resource b_i. In other words,

$$(20) \qquad \lambda_i = \text{imputed value of } b_i = \frac{\partial \Pi}{\partial b_i} \geq 0$$

The dual values can be thought of as the marginal effect on the objective function of a small change in resource availability.

Note: If the constraint is not binding λ_i is always equal to zero, by the Kuhn Tucker Complementary Slackness Conditions. Intuitively, if a resource is not limited then an additional unit is not useful for the production process and the marginal value of that additional unit would be zero. Extra land is of no use (for growing crops) if we don't have any labor!

3.2.2 Dual Objective function

The dual objective function $\boldsymbol{\lambda}'\mathbf{b}$ is equal to the sum of the imputed values of the total resource stock of the firm. Thought of another way, it is the sum of money that you would have to offer a firm owner to make them consent to a buy-out. It is a useful representation of our problem when we are not able to observe primal production data or the production technology.

3.2.3 Dual Constraints

The dual constraints $\mathbf{A}'\boldsymbol{\lambda} \geq \mathbf{c}$ can be interpreted as defining the set of prices $(\boldsymbol{\lambda})$ for the fixed resources (or assets) ($\mathbf{b}$) of the firm that would yield at least an equivalent return to the owner as producing a vector of products ($\mathbf{x}$), which can be sold for prices ($\mathbf{c}$), from these resources. The dual constraint is "greater than or equal to" because you can pay too much for a productive input, but market forces will ensure that you cannot count on buying an input for less than its value when turned into a saleable product.

Post multiplying the transpose of the technical input requirement matrix by the dual prices results in an m x 1 vector of marginal opportunity costs for each of the n potential production activities.

$$(21) \qquad \mathbf{A}'\boldsymbol{\lambda} = \text{vector of marginal opportunity costs of production.}$$

For a single production activity x_i its opportunity cost of production for a vector of dual prices $\boldsymbol{\lambda}$ is:

$$(22) \qquad \mathbf{a}_i'\boldsymbol{\lambda} = (\text{column } i \text{ of } \mathbf{A})'(\boldsymbol{\lambda} \text{ vector})$$

and we note that:

(23) $\mathbf{a}_i'\boldsymbol{\lambda} = \left(a_{ij} \ldots a_{mi}\right)\begin{pmatrix}\lambda_1 \\ \vdots \\ \lambda_m\end{pmatrix}$ = imputed cost of the last unit of x_i produced, which equals the sum of imputed values of each resource needed, multiplied by the amount of that resource needed to produce x_i

(24) $\mathbf{a}_i'\boldsymbol{\lambda}$ = The cost of producing a unit of x_i paying $\boldsymbol{\lambda}$ for resources $\mathbf{b}$

The constraint $\mathbf{A}'\boldsymbol{\lambda} \geq \mathbf{c}$ can be interpreted as saying that the marginal opportunity cost of producing a vector of x_i must be greater than or equal to the marginal revenue ($\mathbf{c}$) for each of the x_i for a maximizing owner of the firm to sell at the firm value of $\boldsymbol{\lambda}'\mathbf{b}$.

4 Positive Mathematical Programming

The Machakos LP example highlights some shortcomings of the LP approach. Most important, the LP has a tendency to overspecialize in crop production. Recall that the optimal solution was to plant the most valuable crop(s) until resource constraints become binding. We know this is not how farmers typically allocate land across a farm (or region) since there is typically variation in land quality and management of pests and disease through crop rotation. We expect, and observe in primal farm and regional data, a diverse mix of crops and livestock. If our model is to be useful for policy evaluation the model should replicate the observed land (and other input) allocation. LP is not able to accomplish this task without introducing a number of restrictive constraints that effectively render the model useless for policy analysis.

One widely used solution is the Positive Mathematical Programming calibration method (Howitt 1995). PMP is a calibration technique that uses observed farmer input and output decisions to calibrate the production model such that the program replicates the base solution. Observed data and economic first-order-conditions are satisfied by the calibrated model which can then be used for policy analysis without restrictive calibration constraints.

We will review calibration theory and the method of PMP in this section of the course.

4.1 Behavioral Calibration Theory

The process of calibrating models to observed outcomes is an integral part of constructing physical and engineering models but is rarely formally analyzed for optimization models in agricultural economics. In this section we show that observed behavioral reactions yield a basis for calibrating models in a formal manner that is consistent with microeconomic theory. Analogously to econometrics, the calibration approach draws a distinction between the two modeling phases of calibration (estimation) and policy prediction.

On a regional level, the information on the product output levels and farm land allocations is usually more accurate than the estimates of marginal crop production costs. This is particularly

true when micro data on land class variability, technology, and risk feature in the farmers' decisions, but are absent in the aggregate cost data available to the model builder. Accordingly, the PMP approach uses the observed acreage allocations and outputs to infer marginal cost conditions for each regional crop allocation observed. This inference is based on parameters that are known to be accurately observed and the usual profit maximizing and concavity assumptions.

The **Nonlinear Calibration Proposition** states: If the model does not calibrate to observed production activities with the full set of general linear constraints it can be empirically justified. A necessary condition for profit maximization at the observed values is that the objective function is nonlinear in at least some of the activities.

Many regional models have some nonlinear terms in the objective function reflecting endogenous price formation or risk specifications. Although it is well known that the addition of nonlinear terms improves the diversity of the optimal solution, there are usually an insufficient number of independent nonlinear terms to accurately calibrate the model.

The **Calibration Dimension Proposition** states: The ability to calibrate the model with complete accuracy depends on the number of nonlinear terms that can be independently calibrated.

The ability to adjust some nonlinear parameters in the objective function, typically the risk aversion coefficient, can improve model calibration. However, if there are insufficient independent nonlinear terms the model cannot be made to calibrate precisely. In technical terms, the number of instruments the modeler has available to calibrate the model may not span the set of activities that need to be calibrated.

For proofs of these latter two propositions, see the appendix of Howitt (1995).

4.2 A Cost Based Approach to PMP Calibration

This section demonstrates the PMP calibration process using nonlinear costs and constant yields to calibrate the model. The key concept of PMP calibration developed in work by Fiacco & McCormack is that every linear constraint in an optimization problem can also be modeled by a nonlinear cost function with appropriately chosen coefficients.

4.2.1 An Initial Primer on Cost Based PMP Calibration – Single Crop

A single linear crop production activity is measured by the hectares x allocated to the crop. Note that we are talking about a single crop in this example. The yield is assumed constant. The data available to the modeler is:

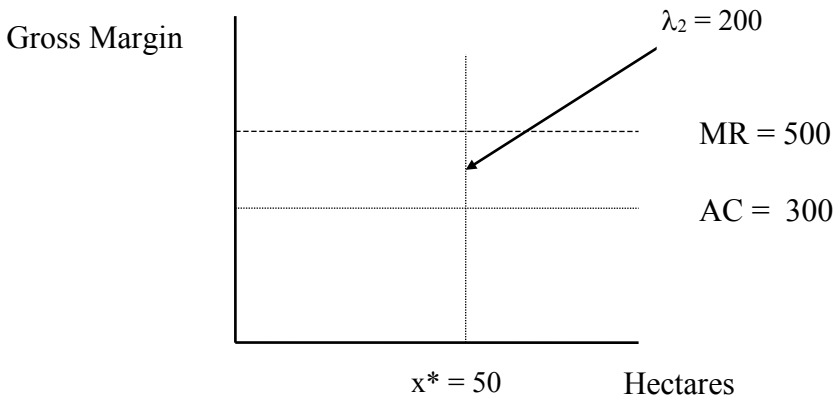
a.	Marginal revenue / hectare is assumed constant at	500 / hectare
b.	Average Cost is	300 / hectare
c.	Observed land allocated to the crop	50 hectares

In the first step we need to estimate the value of the residual cost needed to calibrate the crop to 50 hectares. We do this by setting marginal revenues equal to marginal cost at the observed land use in a constrained linear program. We call this constraint the calibration constraint.

(25) $\text{Max } 500x - 300x$

(26) subject to $x \leq 50$

Illustration of LP Constrained by calibration constraints:



From the nonlinear calibration proposition we know that either (or both) the cost or production function must be nonlinear if we need calibration constraints. In this case we define the total cost function to be quadratic in land (x) of the form:

(27) $TC = \alpha x + 0.5\gamma x^2$

Under optimization, if unconstrained, crop land expands until the marginal cost equals marginal revenue. Therefore we require,

(28) MC = MR at $x = 50$

It follows that the value λ_2 in the linear model is the difference at the constrained calibration value and is equal to MR - AC. But, from Step 2 we know that MR = MC, therefore λ_2 = MC – AC, since MR = MC at x = 50.

Given the hypothesized total cost function TC

(29) $MC = \alpha + \gamma x$

(30) $AC = \alpha + 0.5\gamma x$

And therefore,

(31) $MC - AC = \alpha + \gamma x - (\alpha + 0.5\gamma x)$

Finally,

(32) $\lambda_2 = MC - AC = \frac{1}{2}\gamma x$

And the cost slope coefficient is calculated as:

(33) $\gamma = \frac{2\lambda}{x^*} = (2 * 200)/ 50 = 8$

We can now calculate the value of α using the AC information in the basic data set as:

(34) $300 = a + 0.5 \cdot 8 \cdot 50$

Therefore

(35) $\alpha = 300 - 200 = 100$

Using the values for α and γ the unconstrained quadratic cost problem is:

(36) $Max\,\Pi = 500x - \alpha x - 0.5\gamma x^2$

and

(37) $\frac{\partial \Pi}{\partial x} = 500 - 100 - 8x$

Solving for $\frac{\partial \Pi}{\partial x} = 0$ which implies MR = MC

(38) $8x = 400$

therefore

(39) $x^* = 50$

It is important to note that the unconstrained model calibrates exactly in x and also in Π

Other important features to note include:

- MC = MR at x= 50
- AC = 300 at x= 50
- The cost function has been "tilted"
- Two types of information are used, x* and AC
- The observed x* quantities need to mapped into value space by the LP before it can be used
- The model now reflects the preferences of the decision maker
- The model is unconstrained by calibration constraints for policy analysis

4.2.2 PMP Calibration – Multiple Crops

The previous section showed that if the correct nonlinear parameters are calculated the model will exactly calibrate to the base year values without additional constraints. The problem addressed in this section is to show how the calibrating parameters can be simply and automatically calculated using the minimal data set for a base year LP. The methodology applies to multiple crops.

Given that nonlinear terms in the supply side of the profit function are needed to calibrate a production model, the task is to define the simplest specification that is consistent with the technological basis of agriculture, microeconomic theory and the data base available to the modeler. A highly probable source of nonlinearity in the profit function is due to heterogeneous land quality. This will cause the marginal cost per unit of output to increase as the proportion of a crop in a specific area is increased. This phenomenon, first formalized by Ricardo (Peach 1993), is widely noted by farmers, agronomists, and soil scientists, but often omitted from quantitative production models.

Defining yields per acre as constant and marginal cost as increasing in land allocation is a considerable simplification of the complete production process. Given the applied goal of this "positive" modeling method, the calibration criteria used is not whether the simple production specification is true, but rather, does it capture the essential behavioral response of farmers, and can it be made to work with the restricted data bases and model structures available.

In many LP problems, such as the Yolo and Machakos examples, the objective function coefficients represent the gross margin per unit land faced by the farmer. The coefficient is composed of the product of the average yield, the price per unit output, with the average variable costs deducted. Since we are specifying the yield as constant but the marginal cost as increasing with land in this model, we now have to separate these components.

We maintain our assumption of Leontief production technology. We will assume that land (defined as input i=1) is the binding constraint for calibration and that the grower faces the quadratic total cost function (in land) as described in the previous section. We will assume that we have a total of $j = 1, 2, 3$ inputs and note that the method generalizes to any number. We can write the program that we seek to calibrate as:

$$(40) \qquad Max \sum_i v_i y_i x_i - \left(\alpha_i + 0.5\gamma_i x_i\right) x_i$$

$$(41) \qquad \text{subject to } \mathbf{Ax} = \mathbf{b} \text{ and } \mathbf{x} \geq \mathbf{0}$$

where $a_{i1} = 1$ and $\mathbf{A} = \left(m \ x \ n\right)$ with elements a_{ij}. And x_i is the land allocated to the crop which yields y_i. And α_i and γ_i are respectively the intercept and slope of the cost function per unit land, and w_j is the cost per unit of the jth input. Notice that the land cost enters through the calibrated cost function whereas all other production costs enter linearly.

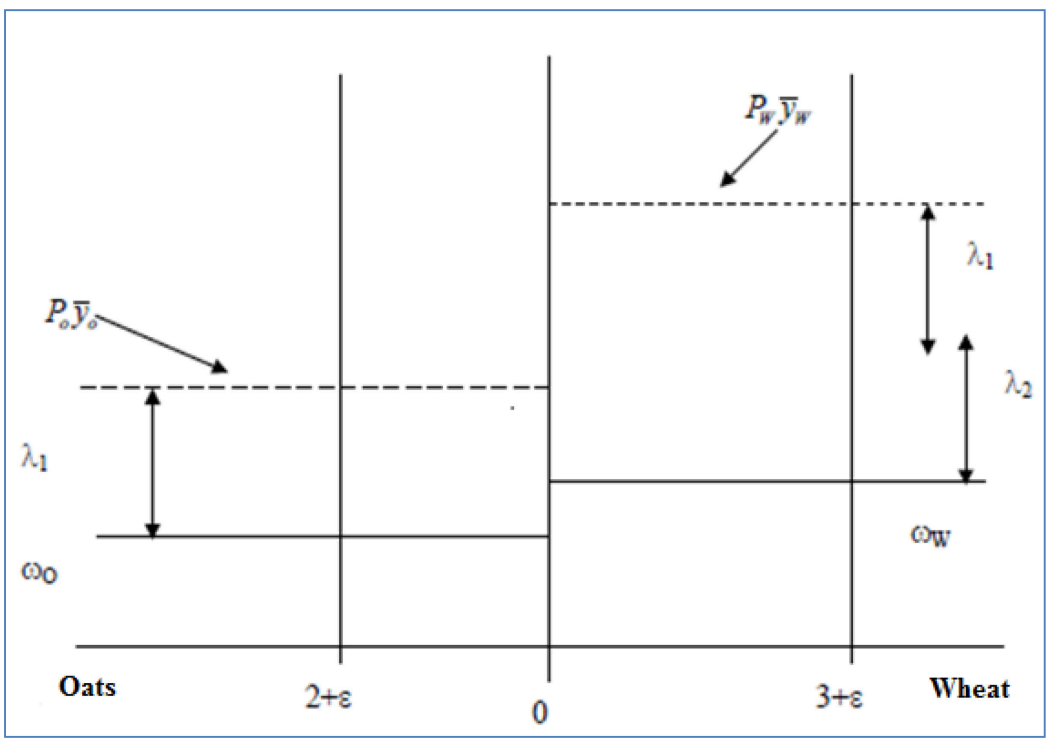
The PMP calibration approach uses three stages. In the first stage a constrained LP model is used to derive the dual values for the resource and calibration constraints, λ_1 and λ_{i2} respectively. In the second stage, the calibrating constraint dual values are used to uniquely derive the calibrating cost function parameters. In the third stage, the cost parameters are used with the base year data to specify the fully calibrated model. The resulting model calibrates exactly to the base year solution and original constraint structure and can then be used for policy evaluation.

The procedure is illustrated using a very simple problem which has a single land constraint (5 hectares) and two crops: wheat and oats. The following parameters are used:

	Wheat (w)	(Oats) (o)
Crop prices	$P_W = V_W = 2.98$/bu.	$P_O = V_O = 2.20$/bu.
Variable cost/hectare	$\omega_W = 129.62$	$\omega_O = 109.98$
Average yield/hectare	$\bar{y}_W = 69$ bu.	$\bar{y}_O = 65.9$ bu.
Gross Margin	76 / ha	35 / ha

The observed land allocation in the base year is 3 hectares of wheat and 2 hectares of oats.

The figure below shows the initial problem in a diagrammatic form for two activities, with one resource constraint and two upper bound calibration constraints. Note that at the optimum, the calibration constraint will be binding for wheat, the activity with the higher average gross margin, while the resource constraint will restrict the land allocated to oats.



The problem in the figure is:

(42) $\max \Pi = (\bar{y}_w v_w - w_w)x_w + (\bar{y}_o v_o - w_o)x_o$

subject to

(43) $x_w + x_o \leq 5 \quad (\lambda_1)$

(44) $x_w \leq 3 + \varepsilon \quad (\lambda_{w2})$

(45) $x_o \leq 2 + \varepsilon \quad (\lambda_{o2})$

Note the addition of the ε perturbation term (0.01) on the right hand side of the calibration constraints. The average gross margin from wheat is 76/ha and oats 35/ha. The optimal solution to the problem is when the wheat calibration constraint is binding at a value of 3.01 and the resource constraint is binding when the oat area equals 1.99 hectares. We will define $\tilde{x}$ as the observed base acres.

Two equations must be solved for the two unknown cost parameters $(a_i \text{ and } \gamma_i)$. This follows the same procedure as the single crop case and we encourage you to verify algebra for yourself.

First, we use the quadratic total land cost function and the first order conditions. We calculate the difference between marginal and average cost and set this equal to the calibration constraint dual value. Note that the difference between the average and marginal cost of land is attributed to changing land quality. We find that,

(46) $MC_i - AC_i = 0.5\gamma_i \tilde{x}_i = \lambda_{i2}$

(47) $\gamma_i = \dfrac{2\lambda_{i2}}{\tilde{x}_i}$

Note the oat calibration constraint is slack, thus $\lambda_{o2} = 0$ and $\gamma_o = 0$.

Next, we can use the average cost with the previously defined γ_i to derive α_i.

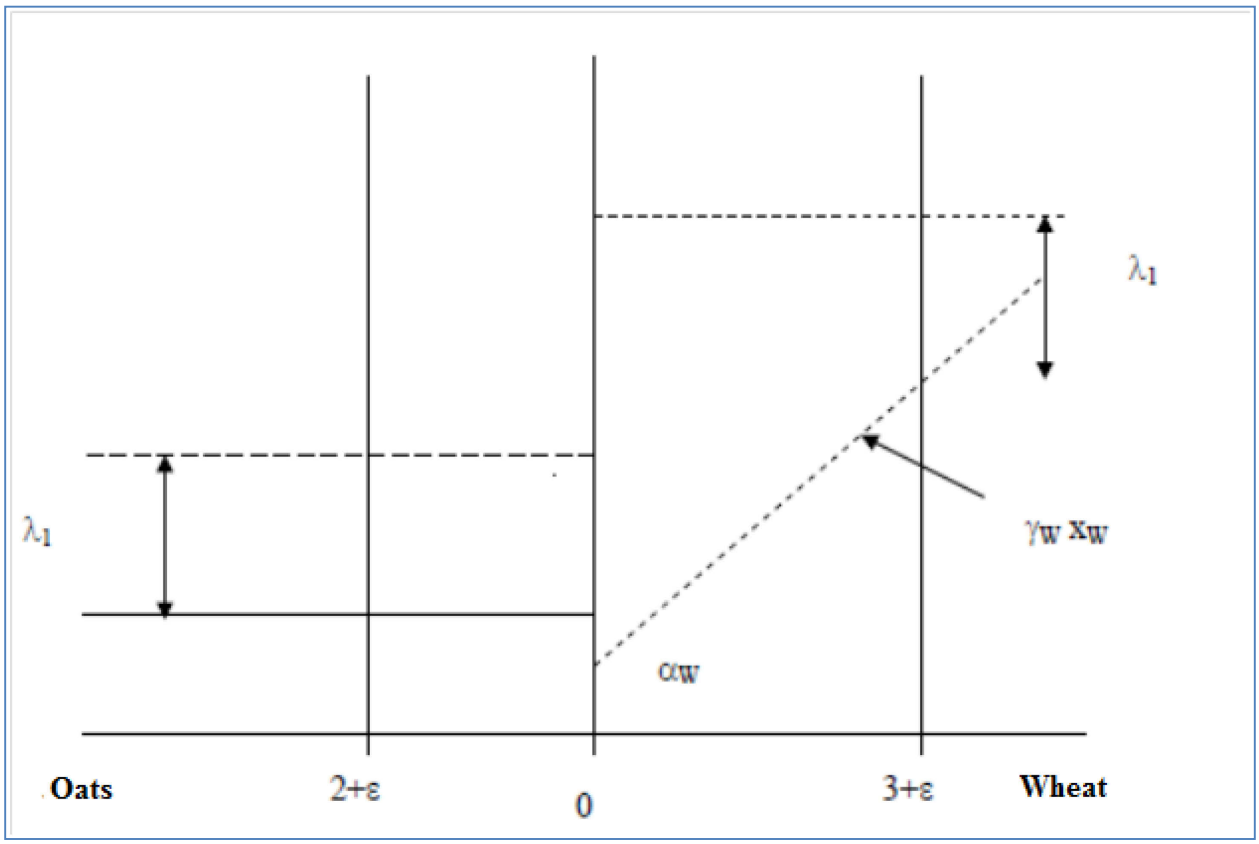
(48) $\alpha_i = AC_i - 0.5\gamma_i x_i$

We can calculate the average production cost from the data. In this case the PMP procedure is a relatively simple calculation that allows us to calculate the parameters of the wheat PMP land cost function. The function allows us to specify our profit maximization problem without restrictive calibration constraints. The figure below illustrates the calibrated problem. Using GAMS we can specify the LP and calculate

(49) $\lambda_1 = 35 \;\; \lambda_{o1} = 0 \;\; \lambda_{w1} = 41$

And it follows that

(50) $\gamma_w = 27.33 \;\; \alpha_w = 88.62$



After we calculate all the cost function parameters we are left with the calibrated optimization problem where we can remove the calibration constraints. For $i = o, w$:

(51) $Max \sum_i v_i y_i x_i - \left(\alpha_i + 0.5\gamma_i x_i\right) x_i \quad where\ i = o, w$

(52) $x_0 + x_w \leq 5$

A quick empirical check of the calibration to the base values is performed by calculating the difference between observed and calibrated land use. We additionally check the PMP cost condition.

The basic PMP model specified calibrates in all aspects. That is, the optimal solution, binding constraints, objective function value and dual values will all be within rounding error of the original LP that is constrained by the calibration constraints. Despite the clunky notation, the basic concept of PMP is numerically simple and easy to solve automatically on a computer.

An alternative calibration approach assumes that the cost per acre is constant, but the yield per acre declines with increased acres to any given crop. It can be shown that this assumption has the

equivalent effect on the profit function as increasing costs. Both specifications can be justified from an agronomic point of view. The cost calibration approach is more common than the yield method. The original PMP article (Howitt 1995) uses a yield-based calibration method, in response to strong feedback from anonymous peer reviewers.

5 Machakos Quadratic PMP Model

We illustrate the mechanics of the PMP procedure using the Machakos data. Here we follow the PMP method outlined in section 4 and solve an LP to get resource and calibration dual values, solve for the parameters of the PMP cost functions, and solve the calibrated non-linear production model.

Follow along with file: Machakos_QuadraticPMP_Day1.gms.

The data are the same as in the Primal LP Machakos model. In the first step we solve the same Primal LP problem, except we add in calibration constraints. The LP:

$$\max \Pi = \sum_i v_i y_i x_i - \sum_{ij} a_{ij} cs_{ij} \tag{53}$$

subject to

$$x_i \le XBASE_i + \varepsilon \quad \forall\ i \tag{54}$$

$$\sum_i x_i \le \sum_i XBASE_i \tag{55}$$

$$x_i \ge 0 \tag{56}$$

We record the dual values from the resource and calibration constraints and analytically calculate the slope and intercept of the PMP cost functions for each crop. Recall,

$$\gamma_i = \frac{2(\lambda_{i2} + adj_i)}{\tilde{x}_i} \tag{57}$$

And

$$\alpha_i = AC_i - 0.5\gamma_i \tilde{x}_i \tag{58}$$

Note the addition of an adjustment term (adj_i) to the calibration dual value. Crops associated with the binding resource constraints will not have calibration constraint dual values. In the absence of prior estimates of the marginal productivity of these crops, we may impose the assumption that marginal productivity decreases 25% over the base condition productivity and thus use 25% of the land resource shadow value as a proxy for the calibration shadow value, and adjust the other calibration values accordingly. While this is a general assumption over different regions

and crops, it provides a robust method for full calibration of all the observed crops without inducing infeasibilities from more arbitrary exogenous restrictions.

At this point we have derived the PMP cost functions that will calibrate the model. In the third stage of the program we combine this information to into the non-linear calibrated program without calibration constraints. The model is fully calibrated. The notes for day 2 will cover how we verify that the model calibrates using a sequential calibration and testing procedure.

$$(59) \qquad Max \sum_i v_i y_i xn_i - \left(\alpha_i + 0.5\gamma_i xn_i\right) xn_i$$

subject to

$$(60) \qquad \sum_i xn_i \leq \sum_i XBASE_i$$

We will investigate further during the afternoon exercises for Day 1, provided with your course materials. We encourage you to work through the GAMS code and verify that the steps for calibration of the Machakos PMP model are as described in the notes.

6 Further Reading

Hazell P.B.R. & R.D Norton "Mathematical Programming for Economic Analysis in Agriculture" Macmillan Co New York, 1986

Howitt R.E. " Positive Mathematical Programming" American Journal of Agricultural Economics 77. (May 1995): 329 -342.

Paris Q. "An Economic Interpretation of Linear Programming" Iowa State University Press, Ames Iowa, 1991.

Howitt and Msangi. "Entropy Estimation of Disaggregate Production Functions: An Application to Northern Mexico." Entropy Journal. 2014.

Paris, Q & R. E. Howitt. "An Analysis of Ill-Posed Production problems using Maximum Entropy" American Journal of Agricultural Economics. 80 (February 1998): pp 124-138.

Fiacco & McCormack. Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Society of Industrial and Applied Mathematics. 1990.

Peach, Terry. Interpreting Ricardo. New York: Cambridge UP, 1993.

Day 2: PMP, Livestock, and CES Production Functions

At the end the day you will be able to:

a. Diagnose and understand whether a PMP model has calibrated
b. Run and interpret a calibrated PMP model with cattle as a production activity.
c. Understand this formulation has a production function which enables us to measure adjustments at the intensive margin.
d. Run and interpret the PMP CES crop model. Calculate the elasticities of supply and input demand.

1 PMP Calibration Checks

Yesterday we saw how to calibrate a crop production model using a limited dataset. We worked through the single crop and two crop PMP calibration examples. How do we know if the model is properly calibrated? There are a few checks that should be performed for any model. We refer the interested reader to the technical treatment in Howitt et al (2012), included with your course material, for further details.

Negative Net Returns

First, we check that our data show positive returns to all the farming activities. This may seem like a trivial check, but it is often the case that we are working with poor or incomplete data that may suggest negative net returns. Intuitively, if there is a negative gross margin associated with a production activity the model will not include that activity in the optimal. When the activity does not enter the solution in the LP we are not able to calculate resource and calibration dual values and therefore are not able to calibrate the model. The check is simply:

$$\mathbf{c} \geq 0 \tag{61}$$

And we will typically use the parameter "NET" in the GAMS program file to denote this test.

LP Difference (DIFF 1)

The second calibration check is the difference in the LP from the baseline observed land use data. We complete this check to ensure that the LP model has reproduced the base year solution, meaning that the resource and calibration constraints have worked as expected. We will typically call this parameter "DIFF1" or "PERDIFF1" in the GAMS code and will calculate it as a percentage difference. A difference of less than 1 percent is generally considered acceptable.

$$(62) \qquad 100 \cdot \left(\frac{\mathbf{x}^* - \mathbf{XBASE}}{\mathbf{XBASE}} \right) \leq \text{tolerance}$$

PMP Check

The third test comes after the calibration of the PMP cost functions. The test is to calculate the difference of the marginal PM cost at the base land allocation from the dual value of the corresponding calibration constraint. A difference of less than a few percentage points is generally acceptable for large scale models.

$$(63) \qquad 100 \cdot \frac{\left(\alpha_i + \gamma_i XBASE_i\right) - \left(\lambda_{i2} + adj_i\right)}{\lambda_{i2} + adj_i} \leq \text{tolerance}$$

VMP Check

For models with a CES production function (discussed later today) we should check that the economic first-order conditions hold. Here we check that the value marginal product of each input equals the cash cost plus opportunity cost of the input. Again, we expect a difference of less than a few percentage points.

$$(64) \qquad 100 \cdot \frac{\left(\text{VMP}_{ij}\right) - \left(w_{ij} + \lambda_{1j} + \lambda_{i2} + adj_i\right)}{\left(w_{ij} + \lambda_{1j} + \lambda_{i2} + adj_i\right)} \leq \text{tolerance}$$

Calibration Check (DIFF 2)

The final test is to check that the calibrated non-linear model reproduces the observed base solution. We will calculate this parameter as percent difference from the base and expect less than a few percent deviation. We use $\mathbf{xn}$ to denote the solution of the non-linear calibrated model.

$$(65) \qquad 100 \cdot \left(\frac{\mathbf{xn}^* - \mathbf{XBASE}}{\mathbf{XBASE}} \right) \leq \text{tolerance}$$

2 Livestock PMP Machakos Model

In this section we will introduce livestock into the crop production model. We will continue with the Machakos model and introduce a livestock activity. Production technology is Leontief and the PMP function is quadratic. Cattle are measured in Tropical Livestock Units (TLU).

Follow along with file: Machakos_Cattle_PMP_Day2.gms.

We have added a production activity which we call cattle and abbreviate CATT. It is defined under the set I. The production activities are now:

(66) $\mathbf{x}' = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]$

(67) $x_6 =$ Cattle

Cattle consume Napier grass which is also produced on the farm. Napier grass is no longer sold off farm for cash and is only fed to and consumed by cattle. Therefore Napier grass is directly linked to the profitability of cattle. To capture this linkage, we introduce a parameter in the matrix of technical coefficients which links Napier grass hay production to cattle feed. We abbreviate this production input as HAYLINK.

As in the Machakos PMP crop model, the LP stage of the Machakos livestock model is used to calculate the resource and calibration dual values. However, we have to add additional constraints to allow for the linked cattle feeding activity.

In the first step we solve the same LP livestock problem, with the addition of calibration constraints. The LP is specified below as:

(68) $$\max \Pi = \sum_i v_i y_i x_i - \sum_{ij} a_{ij} cs_{ij}$$

subject to

(69) $$x_i \leq XBASE_i + \varepsilon \quad \forall \ i$$

(70) $$\sum_i x_i \leq \sum_i XBASE_i$$

(71) $$a_{\text{CATT,HAYLINK}} x_{CATT} + a_{\text{GRASS,HAYLINK}} x_{GRASS} = 0$$

(72) $$x_i \geq 0$$

The third constraint translates Napier grass production on the farm to cattle production (TLU). We specify the model where each TLU requires 0.227 hectares of Napier grass (GRASS). We proceed with the standard PMP procedure and record the dual values from the resource and calibration constraints with one modification. The cattle activity does not have separate resource use and therefore we cannot calibrate this activity in the LP model. Cattle profitability is linked through the Napier grass production activity. Therefore Napier grass has a resource and/or calibration shadow value, but cattle do not.

In the GAMS code we introduce the set “IL” which is a subset of the full crop set “I” that excludes the cattle constraint. You can see that the LP program and PMP procedure are defined over the set “IL” and this allows us to exclude cattle.

Finally we define the calibrated non-linear program.

(73) $$Max \sum_i v_i y_i xn_i - \left(\alpha_i + 0.5\gamma_i xn_i\right) xn_i$$

subject to

(74) $$\sum_i xn_i \leq \sum_i XBASE_i$$

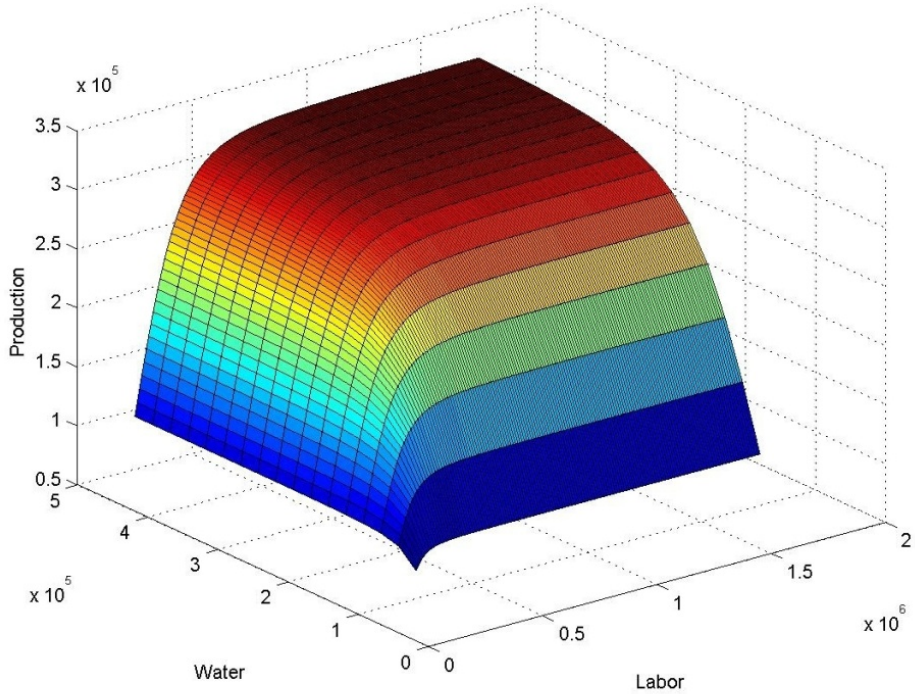
(75) $$a_{\text{CATT,HAYLINK}} xn_{CATT} + a_{\text{GRASS,HAYLINK}} xn_{GRASS} = 0$$

The additional constraint in the non-linear calibrated program is the cattle – Napier grass linkage that we saw in the LP. The same logic holds here: cattle are only produced through Napier grass at a fixed ratio of TLU to hectares of grass production. We can relax this assumption with a more flexible production relationship such as the Constant Elasticity of Substitution (CES) production function. The CES allows for limited substitution between inputs and therefore allows for adjustment along the intensive margin.

3 CES Production Function

A CES production function is a flexible production relationship. An example is shown in the figure below and the theoretical properties may be found in Beattie and Taylor (1985). Notice in the figure below the limited substitutability between inputs in order to produce output.

The figure illustrates two important aspects of the CES production function. First, substitution between inputs is seen by holding production constant (the vertical axis) and sliding around the production surface. The example shown is for the CES production function of a single hay crop in a region with irrigated agriculture. There are many input to production but these are held constant and the figure illustrates the tradeoff between labour and water application. There is limited substitution between water and labour, as shown by the "sharp" corners to the production surface. We also see the ability to model deficit (stress) irrigation by farmers or, more generally, the marginal product of a given input. Faced with a water shortage, farmers may deficit-irrigate some crops. Holding labour constant and sliding along the production surface, as water is decreased, production (yield) decreases as well. Additional restrictions can be imposed to incorporate exogenous agronomic data, when available (Merel, Simon, Yi 2011).



In this section we will see how to calibrate a CES production function. It turns out that although the function is highly non-linear, Howitt (1995) showed the sequential procedure to analytically derive the parameters.

3.1 CES Parameter Calibration

For additional generality we will introduce the regional index g such that we can have a CES production function for each crop in multiple regions. In the Machakos examples we set $g = 1$. CES production function for each region and crop are sequentially derived following the procedure developed in Howitt (1995). The CES is a flexible functional form which allows for a constant rate of substitution between production inputs. The CES production function nests Leontief (fixed proportions) and Cobb-Douglas (unit substitution) production technologies.

We assume Constant Returns to Scale (CRS) CES production functions for every region. This gives us some useful mathematical properties that, when combined with the first-order conditions for an optimum input allocation, yield a sequential set of conditions to solve for the parameters of the CES. We can relax this assumption at the cost of additional complexity (Merel and Bucaram 2010). The CES is defined as:

$$y_{gi} = \tau_{gi}\left[\beta_{gi1}x_{gi1}^{\rho_i} + \beta_{gi2}x_{gi2}^{\rho_i} + \ldots + \beta_{gij}x_{gij}^{\rho_i}\right]^{\upsilon/\rho_i}, \tag{76}$$

where y_{gi} represents output of crop i in tonnes for region g, by combining aggregate inputs j. The scale parameters are τ_{gi}, and the relative use of production factors is represented by the share parameters β_{ij}. The returns to scale coefficient is ν and the CRS assumption requires that the coefficient is equal to1.

The CES is non-nested with the same, fixed elasticity of substitution between any two inputs. If data are available, the substitution elasticity should be estimated. However, if substitution elasticities are available from existing studies, those can be used. We will fix the value equal to $\sigma = 0.7$ for all inputs, allowing for limited substitution between the production inputs. Limited substitution between inputs is consistent with observed farmer production practices. Namely, we observe that farmers can, over a limited range, substitute among inputs in order to achieve the same level of production.

The first order condition for optimal input allocation is that the value marginal product (output price times the marginal product) of each input for each crop and region is equal to the marginal cash cost plus opportunity cost of the input. We have seen this relationship several times in the treatment of the PMP procedure. For inputs this is equal to the base input price plus the dual value on the resource constraint, λ_{1g}, and when binding, the dual value on the calibration constraint, λ_{ig2}.

The parameters of the CES production functions are calibrated using a sequential process. Let the cost per unit of each input, *inclusive* of marginal cash cost and opportunity cost of input j be ω_j. To simplify notation, consider a single crop and region and normalize the price per unit output to be 1. Define

$$\rho = \frac{\sigma - 1}{\sigma}, \tag{77}$$

and the corresponding farm profit maximization problem, optimizing over input use x_j, is written as,

$$\max_{x_j} \pi = v_j \tau \left[\sum_j \beta_j x_j^{\rho} \right]^{\upsilon/\rho} - \sum_j \omega_j x_j. \tag{78}$$

Constant returns to scale requires that $\upsilon = 1$ and thus

$$\sum_j \beta_j = 1. \tag{79}$$

We can use the restrictions imposed by constant returns to scale and take ratios of any two first order conditions to derive the familiar optimality condition that marginal rate of technical

substitution equals the ratio of input costs. Let l correspond to all $j \neq 1$ and rearranging and using the CRS restriction we can explicitly solve for the first (or any arbitrary) coefficient,

$$\beta_1 = \frac{1}{1 + \frac{x_1^{(-1/\sigma)}}{\omega_1}\left(\sum_l \frac{\omega_l}{x_l^{(-1/\sigma)}}\right)}. \tag{80}$$

We use the same procedure as above for all other β_l where $l \neq 1$, thus

$$\beta_l = \frac{1}{1 + \frac{x_1^{(-1/\sigma)}}{\omega_1}\left(\sum_l \frac{\omega_l}{x_l^{(-1/\sigma)}}\right)} \frac{\omega_l}{\omega_1} \frac{x_1^{-1/\sigma}}{x_l^{-1/\sigma}}. \tag{81}$$

We calculate the scale parameter, for each region and crop, from the definition of the CES production function, evaluated at the base level. The scale parameter is

$$\tau = \frac{(yld / \tilde{x}_{land}) \cdot \tilde{x}_{land}}{\left[\sum_j \beta_j x_j^{\rho}\right]^{\upsilon/\rho_i}}. \tag{82}$$

The process generalizes to any number of regions and crops. Using GAMS this process can be automatically and quickly performed for all crops and regions and the production functions will fully calibrated.

3.2 CES and Numerical Scaling

From the first order conditions we see that

$$\beta_l = \frac{\omega_l}{\omega_1} \frac{\beta_1 x_1^{(-1/\sigma)}}{x_l^{(-1/\sigma)}}, \tag{83}$$

for any given input l. If input costs (marginal cash cost plus opportunity cost) of two inputs are of a different order of magnitude this can cause the β_j coefficients to become unbalanced and lead to numerical issues with calibration of the model. Specifically, an ill-conditioned calibration routine will tend to set $\beta_l \approx 1$ and all other $\beta_j \approx 0$. In turn, the model will not calibrate with a low elasticity of substitution (large value in the exponent). This type of data issue is common with large-scale regional production models since inputs are aggregated into coarse categories. For example, other supplies have a much larger cost per unit land than labor costs for many crops causing ill-conditioned matrices that make convergence in solution routines harder and slower to converge to an optimal solution.

There are many sophisticated scaling approaches, but a simple solution that we use is to numerically scale input costs into units of the same order of magnitude. We use land costs as the reference scale and convert input costs, except for land, into land units. We calculate the ratio of input use to total hectares, for each crop and region, and normalize the costs of production into the corresponding unit. This scaling is used throughout the entire program. At the end of the program there is a de-scaling routine which simply reverses this process to convert input use and costs back into standard units.

4 Machakos CES PMP Model

To illustrate the importance of the CES production function we have provided GAMS code to calibrate the Machakos model with a CES production function.

Follow along with file: Machakos_CES_Crops_PMP_Day2.gms.

We specify the model with the same data used in the Primal LP and the Quadratic PMP with Leontief production technology. The model specification is the same as the quadratic PMP model specified in the day 1 notes. We encourage you to review that section of your notes while working through the Machakos CES model.

There is one important change when we move to more complicated models such as the CES. In the models we have seen thus far the calibrated PMP cost function represents all crop production costs in the model. Now we will separate the cost of other inputs from land and use land as the binding resource constraint. We can also imagine situations where other inputs are binding at the farm (for example, labour or water), and the logic readily extends to these situations. Now we define $\lambda_{i2,land}$, where we introduce the land index to represent denote the resource constraint will be binding only for the land input.

In the Machakos CES program we redefine the non-linear profit maximization program with CES production as:

$$(84) \qquad Max \sum_i v_i y_i xn_i - \left(\alpha_i + 0.5\gamma_i xn_i\right) xn_i - \sum_{ij} \omega_i x_{ij}$$

subject to

$$(85) \qquad \sum_i xn_i \leq \sum_i XBASE_i$$

$$(86) \qquad y_i = \tau_i \left[\beta_{i1} x_{i1}^{\rho_i} + \beta_{i2} x_{i2}^{\rho_i} + \ldots + \beta_{ij} x_{ij}^{\rho_i} \right]^{\upsilon/\rho_i}$$

We will investigate the importance of the CES in the afternoon technical exercises. The key feature of the CES is the ability to estimate response along the intensive margin of adjustment. We are able to estimate how farmers may adjust input use intensity in response to new policy or shocks.

5 Calibrating Demands with Limited Data

As a final extension of the Machakos model with CES production function we want to introduce downward sloping demand functions. These are important when the producer(s) may influence the price of a commodity. We will illustrate by way of numerical example.

We often need to derive parameters for demand and supply when the only data available is the equilibrium price and quantity in the base year for the model, and an estimate of the elasticity from a previous econometric study. There is a very simple derivation that enables one to calibrate the slope and intercept coefficients using the elasticity.

Assume that a demand function can be specified as linear in form:

$$(87) \qquad v_i = \text{int}_i + \delta_i y_i$$

Where y_i is the total output of crop i defined from the CES production function. Parameters to be calibrated are int_i and δ_i. Recall that that demand elasticity is defined as:

$$(88) \qquad \eta_i = \frac{\partial y_i}{\partial v_i}\frac{v_i}{y_i}$$

And the price flexibility is $\mu_i = 1/\eta$. Assume we observe the price flexibility of each crop. We can rearrange the flexibility relationship and write the slope:

$$(89) \qquad \delta_i = \frac{\mu_i v_i}{y_i}$$

Given the slope of the demand function we can analytically derive the intercept.

$$(90) \qquad \text{int}_i = v_i - \delta_i y_i$$

We can program a price check to verify that the calibrated demand function equals the observd input price at the observed base level of production. To summarize, given p, y, and an elasticity (or flexibility) estimate, we can derive the corresponding functional form of the demand (or supply) function which, in turn, we can incorporate into the calibrated model. This procedure generalizes to the case of many regions, as shown in Howitt et al (2012).

6 Machakos CES PMP Model – Endogenous Prices

To illustrate the importance of the CES production function and endogenous prices we have provided GAMS code to calibrate the Machakos model. We will not cover this model in this course but encourage you to review on your own and see the reference in Howitt et al (2012) for additional information.

The calibration procedure is unchanged. We now redefine the non-linear profit maximization program with endogenous prices and CES production as:

$$(91) \qquad Max \sum_i v_i y_i xn_i - \left(\alpha_i + 0.5\gamma_i xn_i\right) xn_i - \sum_{ij} \omega_i x_{ij}$$

subject to

$$(92) \qquad \sum_i xn_i \leq \sum_i XBASE_i$$

$$(93) \qquad y_i = \tau_i \left[\beta_{i1} x_{i1}^{\rho_i} + \beta_{i2} x_{i2}^{\rho_i} + \ldots + \beta_{ij} x_{ij}^{\rho_i} \right]^{\upsilon/\rho_i}$$

$$(94) \qquad v_i = \mathrm{int}_i + \delta_i y_i$$

We have introduced the demand function as an additional equation in the model. Alternatively, we can directly incorporate the function in the model objective function.

7 Further Reading

Beattie, B. R., and Taylor, C. R., 1985. The Economics of Production. Krieger Publishing Company.

Howitt, R. E.,1995. Positive Mathematical-Programming. American Journal of Agricultural Economics 77 329-342.

Howitt, Richard E., 1995, A Calibration Method for Agricultural Economic Production Models. Journal of Agricultural Economics. 46(2): 147-159.

Chambers. 1988. Applied Production Analysis: A Dual Approach.

Merel, P., and Bucaram, S., 2010. Exact Calibration of Programming Models of Agricultural Suopply Against Exogenous Supply Elasticities. *European Review of Agricultural Economics* 37 (3) 395-418.

Merel, P., Simon, L., Yi, F., 2011. A Fully Calibrated Generalized CES Programming Model of Agricultural Supply. *American Journal of Agricultural Economics*.

Merel and Howitt. 2014. Theory and Application of Positive Mathematical Programming in Agriculture and the Environment. Review of Agricultural Economics.

Day 3: Dynamic Programming

At the end of the day you will be able to:

a. Understand Bellman's Principle of Optimality and the basic Dynamic programming problem.
b. Have your cake and eat it too.
c. Solve the DP with value function iteration and using Chebychev Polynomial approximation
d. Apply the concepts to a model of Senegal livestock
e. Make changes to the DP and simulate the corresponding change in optimal solution

1 Introduction to Dynamic Programming

Consider the dynamic problem:

$$(1)\ \max f(x_t, c_t)$$

subject to the equation of motion

$$(2)\ x_{t+1} = g(x_t, c_t)$$

Where we define x_t as the state variable which we interpret to describe the "state" of the problem at any time t. For example, this could be seasonal rainfall. Associated with any state variable we have a co-state which you can think of as the dynamic analog of the dual value. We will return to this point later. We define c_t as the control variable at any time t. The control is the action we take in response to the state of the world. For example, this could be the seasonal crop planting decision. The essence of the dynamic problem is to determine a dynamically optimal series of actions (controls) at every time t in response to states of the world.

There are a number of methods for analytically solving the dynamic problem but as we will see these become complicated quickly. In fact, it is rare that we are able to obtain an analytic solution for most real-world applications. Nonetheless, for simple 2 period problems we can proceed with standard optimization methods. In more complicated problems we rely on either the Calculus of Variations (Euler Conditions) or Optimal Control (Pontryagin Conditions). Solution of the dynamic problem requires evaluating a system of differential equations. Analytic solutions are difficult, and in many cases impossible, for most dynamic problems in economics. A third option, which is commonly used for numerical methods, is the Dynamic Programming (DP) approach.

Bellman (1957) developed the fundamental principle of dynamic programming, referred to as the "principle of optimality." The principle states that an optimal sequence of actions has the property that, whatever the initial state and decisions are, the remaining decisions constitute an optimal sequence of actions resulting from the first decision. In other words, whatever the decision maker does tomorrow must be optimal going forward, conditional on the state and decision today. There are a number of solution techniques built around this principle. The most popular include backward recursion, value function iteration, and policy function iteration.

The critical insight by Bellman (1957) allows us to reformulate the dynamic problem using the Bellman equation, and solve using dynamic programming techniques. The Bellman equation for (1) and (2) can be written as:

$$(3)\quad V(x_t) = \max_{c_t} \left\{ f\left(c_t, x_t\right) + \beta V\left(x_{t+1}\right) \middle| \; x_{t+1} = g\left(x_t, c_t\right) \right\}$$

If we allow for an infinite time horizon and let x^+ denote the subsequent period state, we can write the Bellman equation in more compact form as

$$(4)\quad V(x) = \max_{c} \left\{ c^{\alpha} + \beta V\left(x^+\right) \middle| \; x^+ = x - c \right\} = \max_{c} \left\{ c^{\alpha} + \beta V\left(x - c\right) \right\}$$

Where β is the discount factor and V(.) is the value function. The value function represents the maximum utility (profits) for a T-period problem, given all initial starting stocks and conditions. If we can solve this function we have the optimal dynamic solution. This framework extends to situations with stochastic state variables.

Bertsekas (1976) showed that if the discount factor (β) equals less than one then the mapping underlying the Bellman Equation is a strong contraction on Euclidean space. Consequently the Contraction Mapping Theorem guarantees the existence and uniqueness of the solution value function. In other words, the dynamic programming formulation (4) has a theoretically guaranteed unique solution which we can find by solving for V(.) as the fixed point of the Bellman Equation. This is known as value function iteration because the algorithm to find the fixed point, discussed below, iterates on the value function.

We will assume the following for all of our analysis:

a. Value functions that are continuous in the controls and state variables.
b. Value functions that are concave in the controls and state variables.
c. A decision-maker who optimizes the sum of expected discounted value.

1.1 Simple Analytical Case: Cake-Eating

Before we get into more complicated examples we will investigate the cake-eating problem – which will illustrate the key elements of solving a dynamic programming problem. The simple cake-eating problem also allows us to obtain a closed-form analytical solution to the infinite-horizon dynamic programming problem, which becomes more difficult (and even impossible) as

the nature of the objective function and transition equation (i.e. equation of motion) becomes more complex.

Follow along with file: CakeEatingDP_Analytical_Day3.gms.

For the generic formulation of the dynamic programming problem given before:

$$(5)\ \ V(x_t) = \max_{c_t} \left\{ f\left(c_t, x_t\right) + \beta V\left(x_{t+1}\right) \middle| \ x_{t+1} = g\left(x_t, c_t\right) \right\}$$

we can derive two key conditions that must hold at the optimal solution

(1) The first order condition with respect to (wrt) the control variable gives us the "Euler" equation

$$(6)\ \ 0 = f_c\left(c_t, x_t\right) + g_c\left(x_t, c_t\right) \cdot \beta \cdot V_c\left(g\left(x_t, c_t\right)\right)$$

(2) The envelope condition obtained from taking the derivative wrt the state variable gives us the 'Benveniste-Scheinkman' condition:

$$(7)\ \ V_x(x_t) = f_x\left(c_t, x_t\right) + g_x\left(x_t, c_t\right) \cdot \beta \cdot V_x\left(g\left(x_t, c_t\right)\right)$$

We will verify that these conditions hold when we derive the solution for our example problem.

1.1.1 Closed-form analytical example of the Cake-Eating Problem

In order to better understand some of the mechanics of dynamic programming, we choose a simple example that gives us a closed-form solution that we can solve on paper (you will do this!!).

This problem boils down to a non-renewable resource extraction problem, where you try and decide how to optimally 'mine' a finite and fixed resource (that does not regenerate). The solution to this problem will help us to understand some of the key elements of dynamic programming that we will replicate with GAMS, using numerical approximation methods.

The cake-eating problem can be written simply as

$$(8)\ \ V(x_t) = \max_{c_t} \left\{ u\left(c_t\right) + \beta V\left(x_{t+1}\right) \middle| \ x_{t+1} = x_t - c_t \right\}$$

Which we could choose to write in a different way, to show that it is an infinite-horizon problem

$$(9)\ \ U(\mathbf{c}) = \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \quad s.t. \quad x_{t+1} = x_t - c_t$$

Which is the discrete-time version of the "optimal control" problem.

The reader should note that there is a rather strong assumption embedded in this formulation – which gives us analytical convenience, but which has strong implications for behavior. If we

calculate the marginal rate of substitution between the current and future period (two adjacent periods in the time horizon) – we obtain:

$$(10) \qquad MRS_{t,t+1}(\mathbf{c}) = \frac{u'(c_t)}{\beta u'(c_{t+1})}$$

Which suggests that the MRS between two adjacent periods doesn't depend on what came before (i.e. your trade-off between lunch and dinner is independent of what you ate for breakfast). In doing empirical work, you might want to relax this to obtain more plausible behavior – but this is beyond the scope of our treatment.

If we hypothesize a very simple benefit (utility) function such as:

$$(11) \qquad u(c_t) = (c_t)^{\alpha}$$

Then our problem would become

$$(12) \qquad V(x_t) = \max_{c_t} \left\{ (c_t)^{\alpha} + \beta V(x_{t+1}) \,\middle|\, x_{t+1} = x_t - c_t \right\}$$

Which we could drop the time sub-script (t) to obtain:

$$(13) \qquad V(x) = \max_{c} \left\{ c^{\alpha} + \beta V(x^{+}) \,\middle|\, x^{+} = x - c \right\} = \max_{c} \left\{ c^{\alpha} + \beta V(x - c) \right\}$$

1.1.2 Solution by backward recursion

If we solve this problem from the last period (where there's "no tomorrow"),

$$(14) \qquad V_1(x) = \max_{c} \left\{ c^{\alpha} + \beta V_0(x - c) \right\}$$

we would expect that the carry-over value is zero – i.e. $V_0(x^{+}) = 0$. In which case, we have a simple static optimization problem

$$(15) \qquad V(x) = \max_{c} \left\{ (c)^{\alpha} \mid c \le x \right\}$$

For which the obvious solution is: $c^{*} = x$ (you eat all of the cake in the last period). From this solution, the maximized value of the problem defines the carry-over value function for the previous period

$$(16) \qquad V_1(x) = \left(c^{*}\right)^{\alpha} = x^{\alpha}$$

So that in the next-to-last period, where

$$(17) \qquad V_2(x) = \max_{c} \left\{ c^{\alpha} + \beta V_1(x - c) \right\},$$

we can now substitute the value function we solved for so that

$$(18) \qquad V_2(x) = \max_c \left\{ c^\alpha + \beta \cdot (x-c)^\alpha \right\}$$

If we take the first-order condition of this problem (the Euler condition), we get

$$(19) \qquad 0 = \alpha c^{\alpha-1} - \alpha\beta \cdot (x-c)^{\alpha-1},$$

which gives us

$$(20) \qquad c^{\alpha-1} = \beta \cdot (x-c)^{\alpha-1} \rightarrow c = \beta^{\frac{1}{\alpha-1}} \cdot (x-c)$$

From which we can define the optimal consumption as

$$(21) \qquad c^* = \frac{x\beta^{\frac{1}{\alpha-1}}}{1+\beta^{\frac{1}{\alpha-1}}} = \frac{x}{1+\beta^{\frac{1}{1-\alpha}}}$$

Using this solution, we can substitute it back into the Bellman equation in order to obtain the maximized value of the problem (i.e. the value function), as

$$(22) \qquad V_2(x) = \left(c^*\right)^\alpha + \beta V_1\left(x-c^*\right) = \left(\frac{x}{1+\beta^{\frac{1}{1-\alpha}}}\right)^\alpha + \beta\left[x - \frac{x}{1+\beta^{\frac{1}{1-\alpha}}}\right]^\alpha$$

which can be simplified with algebra to

$$(23) \qquad V_2(x) = \left(\frac{x}{1+\beta^{\frac{1}{1-\alpha}}}\right)^\alpha + \beta\left[\frac{x\beta^{\frac{1}{1-\alpha}}}{1+\beta^{\frac{1}{1-\alpha}}}\right]^\alpha = \left(\frac{x}{1+\beta^{\frac{1}{1-\alpha}}}\right)^\alpha \cdot \left[1+\beta\cdot\beta^{\frac{\alpha}{1-\alpha}}\right] = \left(1+\beta^{\frac{1}{1-\alpha}}\right)^{1-\alpha} \cdot x^\alpha$$

If we let $\left(1+\beta^{\frac{1}{1-\alpha}}\right)^{1-\alpha} = \Theta_1$, then our value function can be written as

$$(24) \qquad V_2(x) = \Theta_1 \cdot x^\alpha$$

So in the next stage, we would solve the problem given by the Bellman equation,

$$(25) \qquad V_3(x) = \max_c \left\{ c^\alpha + \beta \cdot \Theta_1 \cdot (x-c)^\alpha \right\}$$

For which we can take the first-order conditions (w.r.t. consumption), to get the Euler condition

$$(26) \qquad 0 = \alpha c^{\alpha-1} - \alpha\beta \cdot \Theta_1 \cdot (x-c)^{\alpha-1} \Rightarrow c^{\alpha-1} = \beta \cdot \Theta_1 \cdot (x-c)^{\alpha-1}$$

And leads to

(27) $$c=\left(\beta\cdot\Theta_1\right)^{\frac{1}{\alpha-1}}\cdot\left(x-c\right)=\frac{x\left(\beta\cdot\Theta_1\right)^{\frac{1}{\alpha-1}}}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{\alpha-1}}}=\frac{x}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}$$

which we can again substitute back into the Bellman equation, in order to obtain the maximized value of the problem as,

(28) $$V_3(x)=\left(c^*\right)^{\alpha}+\beta V_2\left(x-c^*\right)=\left(\frac{x}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}\right)^{\alpha}+\beta\left[x-\frac{x}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}\right]^{\alpha}$$

Which can be simplified (with some algebra) to obtain

(29) $$\begin{aligned}V_3(x)&=\left(\frac{x}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}\right)^{\alpha}+\beta\cdot\Theta_1\left[\frac{x\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}\right]^{\alpha}\\&=\left(\frac{x}{1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}}\right)^{\alpha}\cdot\left[1+\left(\beta\cdot\Theta_1\right)\cdot\left(\beta\cdot\Theta_1\right)^{\frac{\alpha}{1-\alpha}}\right]=\left(1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}\right)^{1-\alpha}\cdot x^{\alpha}\end{aligned}$$

If we let $\left(1+\left(\beta\cdot\Theta_1\right)^{\frac{1}{1-\alpha}}\right)^{1-\alpha}=\Theta_2$, then our value function becomes

(30) $$V_3(x)=\Theta_2\cdot x^{\alpha}$$

Which is of the same functional form as the value function that we got from the previous stage. Based on this pattern – we can see that in each stage (s), the value function takes on the form,

(31) $$V_{s+1}(x)=\Theta_s\cdot x^{\alpha}.$$

Comparing the results we obtained from the previous two stages of the backward recursion, we can see that the parameter that defines the carry-over value function has the following 'equation of motion':

(32) $$\Theta_s=\left[1+\left(\beta\cdot\Theta_{s-1}\right)^{\frac{1}{1-\alpha}}\right]^{1-\alpha}$$

Which we can simulate forward, beginning from a starting value, in order to reach convergence. We can infer that the steady-state value of the parameter would satisfy; $\Theta_s=\Theta_{s-1}=\tilde{\Theta}$. Which we can substitute into the equation of motion

(33) $$\tilde{\Theta}=\left[1+\left(\beta\cdot\tilde{\Theta}\right)^{\frac{1}{1-\alpha}}\right]^{1-\alpha}$$

and then solve for the value of the steady-state parameter. Doing so will give us:

$$(34) \qquad \tilde{\Theta} = \left[1 - \beta^{\frac{1}{1-\alpha}}\right]^{\alpha-1}$$

Which now gives us a steady-state value function,

$$(35) \qquad V_{\infty}(x) = \tilde{\Theta} \cdot x^{\alpha} = \left[1 - \beta^{\frac{1}{1-\alpha}}\right]^{\alpha-1} \cdot x^{\alpha}$$

And which helps us to define the exact form of the Bellman equation, for the infinite-horizon problem as,

$$(36) \qquad V(x) = \max_{c} \left\{ c^{\alpha} + \beta \cdot \left[1 - \beta^{\frac{1}{1-\alpha}}\right]^{\alpha-1} \cdot (x-c)^{\alpha} \right\}$$

So we see that by using a backward-recursion method, we're able to obtain the closed-form solution to the infinite-horizon value function,

$$(37) \qquad V_{\infty}(x) = \left[1 - \beta^{\frac{1}{1-\alpha}}\right]^{\alpha-1} \cdot x^{\alpha},$$

and that we can also obtain (from the Euler equation) the "policy function" that determines consumption as a function of the current state

$$(38) \qquad c = \frac{x}{1 + \left(\beta \cdot \tilde{\Theta}\right)^{\frac{1}{1-\alpha}}} = \frac{x}{1 + \left[\beta \cdot \left(1 - \beta^{\frac{1}{1-\alpha}}\right)^{\alpha-1}\right]^{\frac{1}{1-\alpha}}} = \frac{x}{1 + \frac{\beta^{\frac{1}{1-\alpha}}}{\left(1 - \beta^{\frac{1}{1-\alpha}}\right)}} = x\left(1 - \beta^{\frac{1}{1-\alpha}}\right)$$

This type of 'consumption rule' is derived directly from the closed-form solution of the DP problem, and embodies a feedback response which is characteristic of the 'closed-loop' nature of DP problems. Looking at the consumption rule, we can see that consumption goes down as the agent becomes more patient (i.e. $c \downarrow \quad as \quad \beta \uparrow$), whereas consumption also falls as the stock decreases over time (i.e. $c \downarrow \quad as \quad x \downarrow$). Both of these make intuitive sense. By contrast to this closed-form decision rule of the DP problem, the 'open-loop' form of the optimal control problem,

$$(39) \qquad \max_{\{c_t\}_{t-1}^{\infty}} U(\mathbf{c}) = \max_{\{c_t\}_{t-1}^{\infty}} \left\{ \sum_{t=1}^{\infty} \beta^{t-1} u(c_t) \quad s.t. \quad x_{t+1} = x_t - c_t \right\}$$

implies that you solve for the entire path of consumption from the outset, rather than determining your consumption in each period in response to the current state.

1.1.3 Solution properties of the cake-eating problem

We have seen that we can solve the cake-eating problem using recursive methods (on paper) – but we can also solve it by model-based optimization. We can:

(1) Solve the Bellman equation directly by optimization techniques in GAMS
(2) Or by using the derived "policy function" to calculate consumption over time directly

We will compare the solutions that come from each approach. The 2nd approach can be done in Excel, in fact, if you had the parameter values (α, β) and the beginning level of stock. We should also verify that the first-order and envelope conditions for the cake-eating problem also hold – to make sure our model conforms to theory. This means that for our problem:

$$(40) \qquad V(x) = \max_c \left\{ c^\alpha + \beta \cdot \tilde{\Theta} \cdot (x-c)^\alpha \right\}$$

We should have

(1) The first-order condition with respect to consumption (the Euler equation)

$$(41) \qquad 0 = \alpha c^{\alpha-1} - \alpha \cdot \beta \cdot \tilde{\Theta} \cdot (x-c)^{\alpha-1}$$

(2) The envelope condition with respect to the state variable (i.e. Benveniste-Scheinkman condition):

$$(42) \qquad V_x(x) = \beta \cdot V_x(x) \rightarrow \alpha \cdot \tilde{\Theta} \cdot (x)^{\alpha-1} = \alpha \cdot \beta \cdot \tilde{\Theta} \cdot (x^+)^{\alpha-1} \rightarrow \quad (x)^{\alpha-1} = \beta \cdot (x^+)^{\alpha-1}$$

,

(3) which implies that the following condition holds over the optimal path,

$$(43) \qquad x^+ = x \cdot (\beta)^{\frac{1}{1-\alpha}}$$

We will verify that these hold for our problem.

1.1.4 Exercises for the Cake Eating Problem

To solidify these concepts, you should do the following:

(1) Use a spreadsheet to calculate the optimal path of consumption (using the policy function), and keep track of the stock (updating with the equation of motion $x^+ = x - c$). Use the values ($\alpha = 0.75, \beta = 0.95$)
(2) Verify that the relationship defined by the Benveniste-Scheinkman condition, $x^+ = x \cdot (\beta)^{\frac{1}{1-\alpha}}$, holds for the path of the stock variable that you just calculated above
(3) Use the 'equation of motion' for the value function parameter $\Theta_s = \left[1 + (\beta \cdot \Theta_{s-1})^{\frac{1}{1-\alpha}}\right]^{1-\alpha}$ to simulate the convergence of its value towards the steady-state, infinite horizon value. You can put any (positive) starting value for Θ, actually – but start with using the value we derived from stage 2 of our recursive exercise (i.e. $\Theta_1 = \left(1 + \beta^{\frac{1}{1-\alpha}}\right)^{1-\alpha}$).
(4) Verify the math that was done for our example – and repeat it using a simple utility function for consumption $u(c) = \alpha \log(c)$

2 Value Function Iteration

The value function iteration algorithm solves for the fixed-point of the Bellman Equation. We can describe the steps for value function iteration:

a. Set a convergence tolerance of $\varepsilon = e^{-6}$
b. Make an initial guess, call it **V**, for the value function at each possible state. The Contraction Mapping Theorem guarantees convergence for any starting value.
c. Compute the value function using **V**, call it **TV**.
d. Compute $|\boldsymbol{TV} - \boldsymbol{V}|$
e. Check if $|\boldsymbol{TV} - \boldsymbol{V}| < \varepsilon$
    i. If true, solution has converged.
    ii. If false, update **V** with **TV** and go to (c). Repeat until convergence.

The value function iteration algorithm shown above is relatively straightforward to program and solve. The framework generalizes with stochastic state evolution by simply taking expectations over future periods.

The algorithm can be applied by discretizing the state-space such that we have a discrete approximation of the value function. This method will work for well behaved problems with a small state-space. For more complicated problems we may want to find a continuous approximation to the value function. We can do this by using a Chebychev polynomial approximation.

3 Function Approximation

The Bellman equation that characterizes the optimal solution is known as a functional fixed point equation. These are difficult to solve because they represent an entire function whose domain contains an infinite number of points. Intuitively, it is difficult to solve for a fixed point when we are dealing with an infinite number of points! Fortunately, we can approximate solutions to functional equations using approximation and interpolation methods.

To interpolate we approximate an analytically intractable function f with a "more" computationally tractable function $\hat{f}$. We will call the latter function our approximating function or approximant. We consider approximating functions that can be written as a linear combination of a set of n basis functions $\phi_1,...,\phi_n$ with basis coefficients c_j that we will estimate.

$$\hat{f}(x) = \sum_{j=1}^{n} c_j \phi_j(x) \tag{44}$$

We can use any number of functions as the basis functions but we will focus on a specific subset that have ideal numerical properties. We will select a degree for the basis functions, n. This

leaves us with n undetermined coefficients which we wish to estimate. To solve for these coefficients we require n conditions (equations). To do this we will select a series n of interpolation nodes, x_n. At these nodes we can require that f and $\hat{f}$ are equal,

$$(45) \qquad \hat{f}(x_i) = \sum_{j=1}^{n} c_j \phi_j(x_i) = f(x_i)$$

And this will give us our conditions! We can use more compact matrix notation and write,

$$(46) \qquad \Phi c = y$$

Where

$$(47) \qquad y_i = f(x_i)$$

And an element in the interpolation matrix is the jth basis function evaluated at the ith interpolation node,

$$(48) \qquad \Phi_{ij} = \phi_j(x_i)$$

When viewed this way we can think of interpolation like a curve fitting (regression) exercise. When we have fewer basis functions than function evaluation nodes we can introduce an error and minimize the sum of squared errors.

$$(49) \qquad \varepsilon_i = f(x_i) - \sum_{j=1}^{n} c_j \phi_j(x_i)$$

There is a large literature on optimal selection of both basis functions and basis nodes that is beyond the scope of this course. We will tell you that one method with ideal numerical properties is the Chebychev nodes and Chebychev polynomials. We refer the interested reader to the references at the end of today's course notes.

3.1 Chebychev Nodes

We need to select the points at which we will evaluate our function. We could simply divide the interval evenly. It turns out this is not numerically efficient and we will typically use Chebychev nodes to divide up the interval. Note that in many applications we will want to divide up the interval (which, as we will see, turns out to be the state-space) based on our knowledge of the DP problem. We simply state that the Chebychev nodes are ideal to use. The Chebychev nodes are:

$$(50) \qquad x_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{n-i+0.5}{n}\pi\right), \ \forall i = 1,2,\ldots,n$$

It turns out that the Chebychev nodes are very nearly optimal.

3.2 Chebychev Polynomials

Next, we need to select our basis functions. We could simply use the power functions: $1, x, x^2, x^3, \ldots$ or we could find a basis with better properties. Chebychev polynomials have a number of ideal properties which are discussed in the supplemental reading material for the course and references at the end of this section of the course notes.

A Chebychev polynomial of order *i* is defined over [-1,1], where we can map the interval [a,b] to [-1,1] by

$$(51) \qquad z = \frac{2(x-a)}{(b-a)} - 1$$

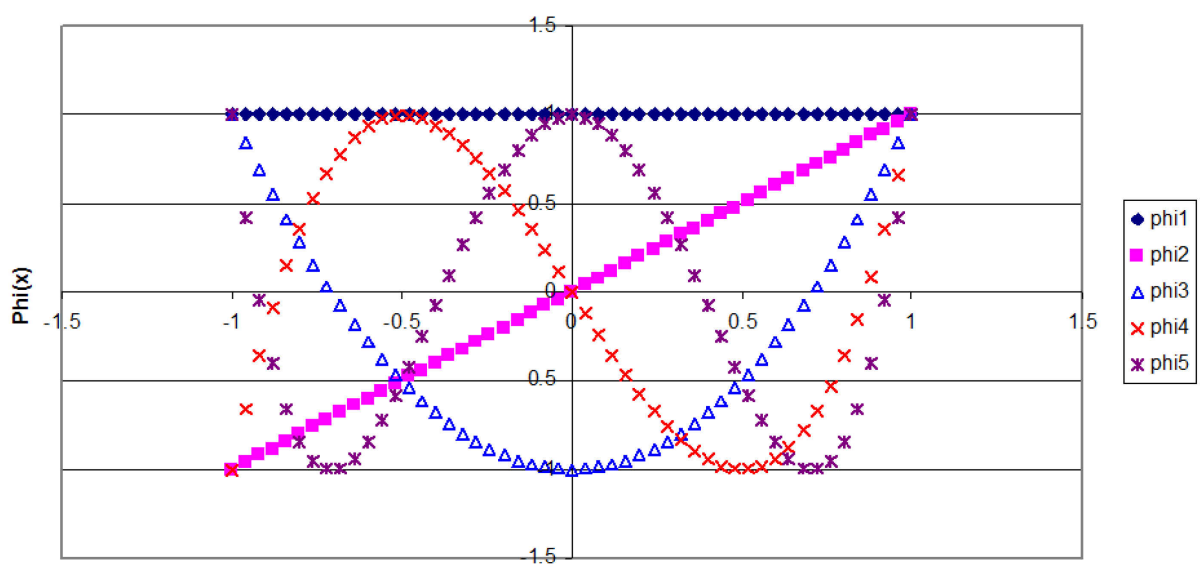
Chebychev polynomial has the following recursive formulation:

$$(52) \qquad T_j(z) = 2zT_{j-1}(z) - T_{i-2}(z)$$

We can see the recursive nature of the polynomial and note that we can evaluate a Chebychev polynomial of order *i* by a simple algorithm:

a. Evaluate the first 2 order polynomials
   i. $T_0(z) = 1$
   ii. $T_1(z) = z$
b. For order i>1 we use these values and the recursive formula
c. Evaluate the resulting polynomial

Below, we show the graph of a Chebychev polynomial with 5 terms (i.e. 5th order), to illustrate how the various polynomials span the [-1, +1] domain



From this we see that the 1st term would be equivalent to an intercept term – which remains invariant over the domain. By adding the 2nd order term, we would obtain a fairly straightforward way of approximating the following type of function: $f(x) = a + bx$. Adding the 3rd order term to the 1st and 2nd would add a curvature similar to that which one would expect from a quadratic function, such as $f(x) = a + bx + cx^2$. Therefore, it is easy to see how adding higher-order terms of a Chebychev polynomial could allow us to approximate functions of increasing complexity (at additional computational cost, of course, since the coefficients for the additional terms must be recovered).

Note that the Chebychev polynomial interpolation matrix has typical element,

$$(53) \qquad \phi_{ij} = \cos\left(\frac{(n-i+0.5)(j-1)\pi}{n}\right)$$

4 "Collocation" or Regression

We can generalize the function approximation methods and show how they can be used to solve a functional equation. In particular, we would like to use these methods to solve the Bellman equation. We can think of the problem as one of finding a function $f:[a,b] \mapsto \mathbb{R}$ that satisfies, for some known function *g,* the relationship

$$(54) \qquad g(x, f(x)) = 0 \;\; for \;\; x \in [a,b]$$

Using our knowledge developed in the approximation section of the notes, we can approximate f using a linear combination of *n* basis functions,

$$(55) \qquad \hat{f}(x_i) = \sum_{j=1}^{n} c_j \phi_j(x_i)$$

And we can estimate the *n* coefficients c_j by requiring that our approximating function satisfies the functional equation at *n* nodes (collocation nodes) in the domain [a,b]. Thus we are left with a numerically tractable problem of finding the *n* coefficients that simultaneously satisfy the *n* nonlinear equations

$$(56) \qquad g\left(x_i, \sum_{j=1}^{n} c_j \phi_j(x_i)\right) = 0 \;\; for \; i = 1, 2, \ldots, n$$

Note that we have replaced a difficult (impossible) infinite-dimension problem with a simpler finite-dimension specification. We will not exactly satisfy the function we are trying to approximate (except, of course, at each of the nodes), thus we will often specify a convergence

criteria. The convergence criteria will allow us to obtain the "best" or at least most acceptable fit of the approximating function.

When $m = n+1$ we call this method collocation and when $m > n+1$ we call this method regression. The latter definition arises because we have more points than the maximum order of the polynomial and we must fit the function by a method such as least squares. The algorithm is the same.

Given our prior discussion, we can see that a good choice of basis function, and possibly, collocation nodes would be Chebychev. Now let's see how to apply this method to the Bellman equation to approximate the value function.

First, define $i = j$ where these are the number of nodes (i) and number of basis functions (j). Then we define the nodes in the -1,1 interval:

$$(57) \qquad x_j = -\cos\left[\frac{\pi(2j-1)}{2n}\right]$$

Next, we define the interpolation matrix :

$$(58) \qquad \begin{aligned} \bar{\phi}_{1,j} &= 1 \\ \bar{\phi}_{2,j} &= x_j \\ \bar{\phi}_{k,j} &= 2x_j\bar{\phi}_{2,j} - \bar{\phi}_{1,j} \quad for\ k \geq 3 \end{aligned}$$

And we define our mapping from map from [-1,1] back to our state space [L,U]:

$$(59) \qquad ST_j = \frac{1}{2}\left[L + U + (U-L)x_j\right]$$

Next, we need to define the value function iteration and coefficient regression as two loops in GAMS.

Inner loop, value function iteration:

Loop over i (possible states current period) using some initial guess for coefficients, say $a_i = 0$. Then solve the DP:

$$(60) \qquad \max CVB_i = u_i + \beta V_i$$

$$(61) \qquad \text{Where, } V_i = \sum_j a_{ij}\phi_{ij}$$

$$\phi_{i,1} = 1$$

(62) $$\phi_{i,2} = \left[\frac{\left(S - \frac{1}{2}(L+U) \right)}{\left(\frac{1}{2}(U-L) \right)} \right]$$

$$\phi_{i,k} = 2\left[\frac{\left(S - \frac{1}{2}(L+U) \right)}{\left(\frac{1}{2}(U-L) \right)} \right] \phi_{i,k-1} - \phi_{i,k-2} \quad for\ k \geq 3$$

And, finally the equation of motion:

(63) $$s_i = s_i - C_i$$

Outer loop, cheby regression (collocation):

Loop over "iter" until convergence of coefficients

(64) Converge when $\sum_i \left(a_i - aOLD_i \right)^2 < tolerance$

(65) Set $aOLD_i = a_i$

(66) Where, $a_i = \dfrac{\sum_j V_j \bar{\phi}_{ij}}{\sum_j \bar{\phi}_{ij}^2}$

In the next section we will see how to apply this algorithm to the cake eating example. Follow along with the GAMS code.

4.1 "Chubby"-chev – Eating Cake

Follow along with file: CakeEatingDP_ChebyApprox_Day3.gms.

If we return to our relatively simple formulation of the cake-eating dynamic programming problem, where we have the general problem

(67) $$V(x_t) = \max_{c_t} \left\{ f\left(c_t, x_t\right) + \beta V\left(x_{t+1}\right) \middle| \ x_{t+1} = g\left(x_t, c_t\right) \right\}$$

Which can be stated in simpler terms

(68) $$V(x) = \max_c \left\{ c^\alpha + \beta V\left(x^+\right) \middle| \ x^+ = x - c \right\} = \max_c \left\{ c^\alpha + \beta V\left(x - c\right) \right\}$$

Rather than solving this by the backward recursion method (which is computationally laborious, as we discovered) – we might choose to undertake a numerical solution by our collocation

method. This will become necessary if we want to deal with a DP problem that is any more complex (even slightly) than the one we have for this problem. The recursion method that we undertook by hand calculations is called 'value function iteration' – because we were solving iteratively for the parameters of the value function that made the maximization of the non-linear programming problem on the RHS of the Bellman equation equal to the value function (expressed in terms of the current value of the state variable) on the LHS of the equation.

Value function iteration can be carried out numerically for our cake eating empirical example, by first defining a polynomial (Chebychev!) to approximate the value function over a range of nodes (maybe Chebychev, maybe other nodes in the state space!), and then successively using the numerical solution values of the maximized Bellman equation to estimate the parameters of the value function approximation. The updated parameters (and value function approximation) can then be used in a subsequent optimization of the updated Bellman equation – such that we iteratively converge upon a set of polynomial values that are consistent with maximized value of the Bellman equation.

If we express our polynomial approximation of the 'true' value function in terms of a convex sum of coefficient and polynomial terms, such as

$$\tilde{V}(x) = \sum_j a_j \phi_j \left(M\left(x\right) \right) \tag{69}$$

where j refers to the order of the polynomial, and we have redefined the coefficients (previously c_j) as a_j to avoid confusion with the control variable c_t in the cake eating problem. The values of each polynomial term $\phi_j(\cdot)$ are evaluated with respect to the state variable (x) – whose value is mapped onto the domain of the polynomial term with the mapping $M(x)$. So if we have a 3rd order polynomial, then the Chebychev polynomial approximation of the value function evaluated at any point in the domain of the state variable (x) can be written out as

$$\tilde{V}(x) = \sum_{j=3} a_j \Phi_j \left(M\left(x\right) \right) = a_1 \phi_1 \left(M\left(x\right) \right) + a_2 \phi_2 \left(M\left(x\right) \right) + a_3 \phi_3 \left(M\left(x\right) \right) \tag{70}$$

For example, we chose 3 points $\left(x_1, x_2, x_3\right)$ over the domain of the state variable – which we can define as the interval (inclusive of the endpoints) $\left[x^{low}, x^{up}\right]$. Note that here we are selecting our nodes and not using the Chebychev nodes discussed earlier. We can then solve the Bellman equation for these current-period values of the state variable and we would obtain a sequence of maximized values such that

$$\begin{aligned} \tilde{V}(x_1) &= \max_c \left\{ c^\alpha + \beta \tilde{V}\left(x_1 - c\right) \right\} = v_1 \\ \tilde{V}(x_2) &= \max_c \left\{ c^\alpha + \beta \tilde{V}\left(x_2 - c\right) \right\} = v_2 \\ \tilde{V}(x_3) &= \max_c \left\{ c^\alpha + \beta \tilde{V}\left(x_3 - c\right) \right\} = v_3 \end{aligned} \tag{71}$$

Where each solution yields a single scalar numerical value $\left(v_{1,2,3}\right)$. This allows us to equate the terms of the polynomial approximation at these computation 'nodes' in the following way

$$\begin{aligned} a_1\phi_1\left(M\left(x_1\right)\right)+a_2\phi_2\left(M\left(x_1\right)\right)+a_3\phi_3\left(M\left(x_1\right)\right)&=v_1 \\ a_1\phi_1\left(M\left(x_2\right)\right)+a_2\phi_2\left(M\left(x_2\right)\right)+a_3\phi_3\left(M\left(x_2\right)\right)&=v_2 \\ a_1\phi_1\left(M\left(x_3\right)\right)+a_2\phi_2\left(M\left(x_3\right)\right)+a_3\phi_3\left(M\left(x_3\right)\right)&=v_3 \end{aligned} \tag{72}$$

We should recognize this as a linear system and we can write the system as

$$\begin{bmatrix} \phi_1\left(M\left(x_1\right)\right) & \phi_2\left(M\left(x_1\right)\right) & \phi_3\left(M\left(x_1\right)\right) \\ \phi_1\left(M\left(x_2\right)\right) & \phi_2\left(M\left(x_2\right)\right) & \phi_3\left(M\left(x_2\right)\right) \\ \phi_1\left(M\left(x_3\right)\right) & \phi_2\left(M\left(x_3\right)\right) & \phi_3\left(M\left(x_3\right)\right) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \tag{73}$$

Which (dropping some notation) we can express as

$$\begin{bmatrix} \phi_{11} & \phi_{21} & \phi_{31} \\ \phi_{12} & \phi_{22} & \phi_{32} \\ \phi_{13} & \phi_{23} & \phi_{33} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \quad \text{or } \mathbf{\Phi a} = \mathbf{v} \tag{74}$$

This should be familiar to you as the same interpolation (or collocation) problem we discussed in the previous section! We effectively have an implicit function which we can approximate by choosing a proper set of basis functions. We can solve for the vector of polynomial coefficients ($\mathbf{a}$) by solving the simple inverse problem $\mathbf{a} = \mathbf{\Phi}^{-1}\mathbf{v}$. This is precisely what is done in the GAMS code that we have for our numerical solution of the deterministic dynamic programming problem – and which you will explore further in the exercises.

4.1.1 Polynomial Approximation Exercises

To solidify these concepts, you should do the following:

1. Write out the linear system which corresponds to a numerical solution of the DP problem, using a 2nd order polynomial that is evaluated at only 2 'nodes' – to convince yourself that you know what it would look like (both the linear system and the inverse solution that yields the coefficient values).
2. Keeping the 2-dimensional formulation, above – hypothesize a value function polynomial approximation of the form: $\tilde{V}(x) = a_1 + a_2 \cdot x$ (which is probably not a very good approximation – but is simple for us to understand).
    a. Write out what the linear system $\mathbf{\Phi a} = \mathbf{v}$ would look like, for this functional form, if you were to solve the Bellman equation at 2 'nodes' (x_1 and x_2)

3. Try solving the Bellman equation using this very simple (linear) polynomial approximation, for the cake-eating problem $V(x) = \max_c \left\{ c^\alpha + \beta V\left(x - c\right) \right\}$, taking $\alpha = \beta = \frac{1}{2}$ and starting off with $a_1 = a_2 = 0$ and evaluating at the nodes $\left(x_1 = 0, x_2 = 9\right)$. Do a few iterations (calculating the value of the maximized problem by hand), until you get a feel for how the coefficient values (a_1, a_2) might change with each successive iteration. This will give you a good appreciation (if you don't already have it) of the computational drudgery that GAMS is saving you from!!

5 Application to Senegal Livestock

We use as an example the Ferlo region of Senegal livestock stocking model presented by Hein (2009), included with your course reading materials. We encourage you to read this paper carefully as the model code closely follows the developments in the paper, with some modification and simplifications discussed in the notes.

Follow along with file: Senegal_LiveStocking_Day3.gms.

The model developed is for Ferlo semi-arid rangeland in northern Senegal. The decision (control) variable in the model is the long-term stocking density. The model accounts for stochastic states in the form of rainfall and the feedback effect of grazing on vegetation. The model abstracts away from common property problems endemic to grazing and should consequently be viewed from the point of the optimal rangeland manager.

The control variable is livestock sold at time t , SL_t. The state is the size of the livestock herd measured in Tropical Livestock Units (TLU), TLU_t. Livestock feed on fodder F_t which is produced on the land depending on rain r_t and Rainfall Use Efficiency (RUE) RUE_t.

Current period profits are defined as

$$\pi_t = SL_t\left(\alpha_0 - \alpha_1 SL_t\right) \tag{75}$$

Where α_0 and α_1 are the intercept and slope of the livestock market demand function, respectively. These parameters are assumed known and fixed. Livestock production is governed by grass feed and resource availability.

RUE defines the efficiency of rainfall expressed in biomass/ha per year per mm effective rainfall. This indicates the effectiveness to transfer rain to biomass.

$$RUE_t = \left(\alpha_2 r_t^2 + 2\alpha_3 r_t - \alpha_4\right) - \left(SR_t\left(\mu r_t^2 - 2\mu R r_t + v\right)\right) \tag{76}$$

Where we have simplified the equation on page 140 (equation 6.1) in the Ferlo paper. Parameters α, μ, v are scaling parameters. The parameter R is the average rainfall (fixed and known).

Finally, SR_t is the long term stocking density calculated as,

$$(77) \qquad SR_t = \frac{TLU_t}{H}$$

Where H is the land area being grazed. The stocking density represents the intensiveness of grazing but does not capture the important variation in spatial distribution of grazing.

Fodder production is governed by the product of RUE and land area. In this model RUE captures all the essential productivity features of the land.

$$(78) \qquad F_t = RUE_t H$$

There is a limit to the production of the land which depends on the feeding requirements of a TLU, assumed fixed and known at 2,500 kg/ha in this model.

$$(79) \qquad TLU_t^{MAX} = \frac{F_t}{Ph}$$

Finally, TLU changes depending on what is sold plus a growth function. Growth is assumed to follow a logistic functional form with shape parameter λ,

$$(80) \qquad TLU_{t+1} = \lambda\left(1 - \frac{TLU_t}{TLU_t^{MAX}}\right)TLU_t - SL_t$$

5.1 DP Solution

We solve the model using the Chebychev polynomial approximation of the value function.

First, we define 4 nodes at which to evaluate the value function approximation. We define the state space (TLU) to be $[20, 50]$ and map to the [-1,1] interval by

$$(81) \qquad \hat{x}_j = \cos\left(\pi \frac{2j-1}{2n}\right), \text{ for } j = 1, \ldots, n$$

The transformation back to the $[20, 50] \equiv [L, U]$ interval can be calculated as

$$(82) \qquad x_j = \frac{\hat{x}_j\left(L + U\left(U - L\right)\right)}{2}$$

Given this mapping we can now define our interpolation matrix using the recursive formula

$$
\begin{aligned}
&\phi_{1,j} = 1 \\
(83) \qquad &\phi_{2,j} = \hat{x}_j \\
&\phi_{k,j} = 2\hat{x}_j \phi_{k-1,j} - \phi_{k-2,j} \quad \forall k \geq 3
\end{aligned}
$$

5.1.1 Value Function Iteration and Chebychev Collocation

We can now solve the Senegal DP following the methods outlined in earlier sections of the notes. We first specify an initial guess (say 0) and solve the DP by value function iteration. The solution allows us to update the approximation polynomial coefficients and we iterate until these converge within acceptable tolerance.

5.2 DP Simulation

Finally we want to use the solution of the DP to simulate the optimal decision sequence and possibly investigate changes in key parameters. We specify the parameters of the optimized DP problem and then solve the optimal time-path forward. The result at each stage is the optimal decision of the agent, in this case the optimal herd stocking decision. Initial values and the evolution of state variables may be specified from existing data or other sources.

6 Further Reading

Kenneth Judd. 1998. Numerical Methods in Economics. The MIT Press.

Dixit and Pindyck. 1994. Investment Under Uncertainty. Princeton University Press.

Adda and Cooper. 2003. Dynamic Economics. The MIT Press.

Valdivia, R, John Antle, Jeste Stoorvogel. In Search of Sustainable Development: Modeling Semi-Subsistence Crop-Livestock Systems to Solve the Poverty-Productivity-Sustainability Puzzle in Sub-Sarahan Africa. Working Paper Oregon State University. December 2013.

Day 4: Stochastic Dynamic Programming

At the end of the day you will be able to:

a. Understand Bellman's Principle of Optimality and the basic Stochastic Dynamic programming problem.
b. Solve the SDP with value function iteration
c. Apply the concepts to models of agro-forestry and livestock herd dynamics
d. Make changes to the SDP and simulate the corresponding change in optimal solution

1 Introduction to Stochastic Dynamic Programming

Today we will extend our DP framework to the (more realistic) situation where we have stochastic state variables in the model. We will then explore multi-state models in the context of herd dynamics and agro forestry management.

We return to the cake eating example to introduce stochastic elements in the dynamic solution, then turn attention to an agro-forestry model.

1.1 Stochastic Cake Eating

We can extend the cake eating example from the deterministic to a stochastic application. We previously assumed that the person eating cake had a known appetite in every period. In other words, the agent's utility was increasing in the amount of cake and this amount was fixed and known. In practice the agent likely has a stochastic appetite, some days you want to eat cake and some days you don't.

Follow along with file: CakeEatingDP_ChebyApprox_Stochastic_Day4.gms.

We can effectively represent stochastic appetite as a shock to tastes and preferences in the current period. Consider a stochastic taste shock ε such that utility from consumption of cake is now $u(c,\varepsilon)$. We will assume that the agent knows that value of the stochastic shock when making a current period decision, but does not know the shock for future periods. Intuitively, we are modeling a situation where you know how much you would enjoy cake today but not tomorrow. The stochastic shock means that the agent should factor in the *potential* future shocks when making current period consumption decisions. As such, we expect that the optimal dynamic solution changes. We will see this.

First we need to define the nature of the stochastic shock. We could assume any type of distribution of the random variable ε. One common assumption is to choose a simple specification by assuming a first-order Markov process. A Markov process can have a number of useful numerical properties. Most important, the probability of future shocks is completely

described by the current period. Let there be two sates, l and h, described by ε_h and ε_l. The transition between states follows a first-order Markov process and can described by the transition matrix Π. An element in the transition matrix tells us the probability of moving from state *i* to state *j* in the next period.

$$(1)\ \Pi = \begin{bmatrix} \pi_{ll} & \pi_{lh} \\ \pi_{hl} & \pi_{hh} \end{bmatrix}$$

$$(2)\ \pi_{ij} \equiv \Pr\left(\varepsilon_{t+1} = \varepsilon_j \mid \varepsilon_t = \varepsilon_i\right)$$

In this case we include the stochastic taste shock in our decision process. The agents choice of how much cake to eat now depends on both the size of the cake and the realization of the taste shock. The agent knows the current shock and how that is expected to transition to future periods. The stochastic cake-eating problem can now be written as

$$(3)\ V(x_t, \varepsilon_t) = \max_{c_t} \left\{ u\left(c_t, \varepsilon_t\right) + \beta E_{\varepsilon_{t+1}|\varepsilon_t} V\left(x_{t+1}, \varepsilon_{t+1}\right) \middle| \ x_{t+1} = x_t - c_t, \varepsilon \sim Markov \right\}$$

The Markov process for the evolution of the taste shock says that today's preferences tell us the probability of tomorrow's preferences. The agent observes his tastes today and knows in a probabilistic sense through the transition matrix how those preferences will evolve in the future. In some cases this representation may not make sense since we may be interested in a stochastic shock (or state) whose realization tomorrow does not depend on the value today. For example, if our taste for cake is not related to our cake consumption in the previous period, or if rainfall next week is not related to rainfall today. Fortunately we can specify any type of random variable in the SDP problem. Consider the case where we specify the taste shock as a different random variable.

Define *e* points at which we know the probability, pr_e, of shock a shock of magnitude shk_e. We will define the probabilities such that

$$(4)\ \sum_e pr_e = 1$$

We can write the stochastic cake-eating problem as:

$$(5)\ V(x_t, e_t) = \max_{c_t} \left\{ u\left(c_t, e_t\right) + \beta E_{e_{t+1}|e_t} V\left(x_{t+1}, e_{t+1}\right) \middle| \ x_{t+1} = x_t - c_t, e \sim RV \right\}$$

Let us assume that the stochastic shock affects utility multiplicatively. Given our definition of the stochastic shock, we can alternatively write,

$$(6)\ V(x_t, e_t) = \max_{c_t} \left\{ shk(e_t)u(c_t) + \beta \sum_e pr(e) shk(e_{t+1}) V\left(x_{t+1}, e_{t+1}\right) \middle| \ x_{t+1} = x_t - c_t \right\}$$

Notice that we have assumed a simple stochastic process where the distribution of *e* in subsequent periods is independent of the current period (as was the case with Markov) and independent of (potentially) other states and the control. The contraction mapping theorem holds for this problem which means there exists a fixed point of the functional equation (Bellman). We can solve for this point using the same methods for the deterministic DP. The only modification is that we now take expectations over future expectations of the stochastic shock.

2 Multi-State Models

The SDP and DP framework both extend naturally to models with several state variables. Most situations where we want to apply the DP or SDP framework will involve multiple states that we need to simultaneously model. For example, the herd stocking decision depends on factors such as prices, disease, and rainfall in addition to the size of the herd and basic population dynamics. In general, we can write for any number of states *m:*

$$(7)\quad V(x_t) = \max_{c_t}\left\{ f\left(c_t, x_t^1, \ldots, x_t^m\right) + \beta V\left(x_{t+1}, \ldots, x_{t+1}^m\right) \middle| \; x_{t+1}^m = g^m\left(x_t^m, c_t\right)\right\}$$

Unfortunately the extension of the dynamic framework to many states is not without cost. As we increase either/or the number of states and/or the dimension of the states the computational cost of the problem increases exponentially. Intuitively we are searching for an optimal decision rule for every possible combination of realized state values. As the number of states increases so does the number of points at which we must evaluate and solve the DP. Even with modern computational power this is a significant limiting factor and is known as the "curse of dimensionality."

2.1 Function Approximation

The methods for function approximation extend naturally to multi-state applications. In particular, we can use the Chebychev approximation approach. Consider the extension to *m* states. First, we define upper and lower bounds for the state variables, $\left[L^m, U^m\right]$ and we can map the empirical state variable values into the [-1,1] interval using the same formula:

$$(8)\quad \hat{x}_j^m = \cos\left(\pi \frac{2j-1}{2n}\right), \text{ for } j = 1, \ldots, n$$

The transformation back to the $\left[L^m, U^m\right]$ interval can be calculated as

$$(9)\quad x_j^m = \frac{\hat{x}_j^m\left(L^m + U^m\left(U^m - L^m\right)\right)}{2}$$

Given this mapping we can now define the Chebychev interpolation matrix using the recursive formula

$$
\begin{aligned}
&\phi_1^m = 1 \\
(10) \qquad &\phi_2^m = \hat{x} \\
&\phi_j^m = 2\hat{x}\phi_{j-1} - \phi_{j-2} \quad \forall j \geq 3
\end{aligned}
$$

Now we have define the state space and Chebychev nodes and basis functions for each state variable m. We can write the Chebychev approximation to the value function as

$$
(11) \qquad V = \sum_{j^1} \ldots \sum_{j^m} a_{j^1 \ldots j^m} \prod_m \phi_j^m
$$

The approximation of the value function with multiple states simply extends the Chebychev polynomials to additional dimensions to approximate the solution over each of the additional states. We will see how to apply this approach in two applications.

3 Agro-Forestry Application

We consider an example of orchard management to illustrate the agro-forestry application. The decision problem facing the agent is when to replant and retire trees in the orchard. There is an establishment cost to new trees which includes the variable production costs plus the amortized fixed costs associated with new plantings

Follow along with file: AgroForestryModel_DP_Day4.gms.

The orchard has trees of three age profiles where each age profile has a different expected yield and therefore expected profitability. Young trees do not typically yield over the first few years, then they mature and then finally decline as they get old. The decision facing the agent is how to manage a fixed amount of land and plant new trees (with a cost) by retiring old trees that have lower average yields. Tress transition between age profiles following a known and time-invariant transition process.

3.1 Input Data and State Space

We simulate the problem over a 20 year time horizon. There are three possible age profiles of the trees: early, mature and old. These are three state variables in the model which correspond to the different age profiles of the trees. The transition between age profiles follows the transition matrix shown below. In any year 60 percent of the early tree plantings transition to mature trees and 30 percent of mature trees transition to old trees.

Transition Matrix	Early	Mature	Old
Early	0.4	0.6	0
Mature	0	0.7	0.3
Old	0	0	1

There is a maximum land area available to the agent of 100 hectares. The cost of uprooting old trees is 20/ha and the cost of replanting is 100. We assume an agent acting to maximize the present discounted value of an infinite stream of profits with discount rate of 5%. The table below summarizes the key parameters in the model. Once you are comfortable with the model code we encourage you to change the key parameters to see how the optimal solution changes.

Model Data	Early	Mature	Old
Price per kg	10	10	10
Yield (kg/ha)	0	10	5
Initial profile (plantings)	10	5	4

3.2 Simulation

The model has three state variables (early, mature, old) and we approximate the solution of the infinite horizon problem by Chebychev approximation of the value function. Following the discussion in the previous section we define m=3 and

$$V = \sum_{j^1} \ldots \sum_{j^m} a_{j^1 \ldots j^m} \prod_m \phi_j^m \tag{12}$$

To speed up convergence we have included code for the initial values of the Chebychev coefficients based on earlier runs. This greatly speeds up convergence time.

4 Herd Dynamics Application

Yesterday we specified a DP model of the stocking decision of livestock management. Today we will specify an example of herd dynamics. The additional complexity here is that as livestock age they have corresponding differences in productivity.

Follow along with file: HerdDynamics_DP_Day4.gms.

There are three state variables in the model, juvenile, female adult, and male adult livestock. Juveniles will transition into either productive males or females with some known and time-invariant transition process. Productive output from the herd includes milk and meat. There is a

limited amount of land for grazing with a fixed and known productivity. The agent faces known and time-invariant market demand for output as well as feed and animal input supply functions.

Females produce milk and new livestock (male and female). Males are sold exclusively for meat but a minimum number are required for breeding purposes. The productivity of adult males and females is fixed and known. The decision facing the agent is when to add to the herd or sell from the herd given all market conditions and resource constraints.

4.1 Input Data

We simulate the model over a 40 year horizon where the discount rate is 5%. The table below summarizes the key input in the model.

Input Data	Juvenile	Adult Male	Adult Female
Animal weight	40	300	275
Milk yield (kg/yr/animal)	0	0	50
Initial animals	60	20	30
Birth rate per female (animal/yr)	1.5	0	0

Females give birth to 1.5 juveniles per year (or relevant period). Juveniles then transition into either male or female adults according to the following transition matrix. In other words, 30 percent of juveniles remain, 30 percent transition to males and 40 percent transition to females.

Transition Matrix	Juvenile	Adult Male	Adult Female
Juvenile	0.3	0.3	0.4
Adult Male	0	1	0
Adult Female	0	0	1

Finally, the herd can be fed through grazing on a fixed amount of land or additional off-farm feed can be purchased. These food sources have different nutrient content and therefore different effects on animal productivity.

4.2 Simulation

We simulate the model over a 100 year time horizon and approximate the value function at 3 Chebychev nodes. The agent acts to maximize the present discounted value of a future stream of profits by selecting, in each period, the animals sold and purchased, as well as milk sold. The agent may purchase off-farm feed and responds to fixed and known market demand and supply for inputs and outputs. The age profile of the herd evolves endogenously according to model parameters.

5 Application to Groundwater Dynamic Optimization

We first apply the Chebychev approximation approach to an empirical example of optimal management of groundwater. This model will not be covered in the course lecture or GAMS exercises, but it should illustrate some of the key points related to solving SDPs.

Groundwater is pumped optimally if it maximizes the expected present value of current and future returns from the water. A decision to pump groundwater today has to consider not only the current marginal net revenue from the water, but also the future net value of groundwater if it remains underground, given the current level of groundwater, the increased cost of pumping if it is reduced, and the probability distribution of future recharge. This can be expressed mathematically as:

$$\max_{d_t} f\left(d_t\right) + \sum_{t+1}^{\infty} \beta_t \sum_i P_i f\left(d_{t+1}, H_{t+1}\right) \tag{13}$$

subject to

$$H_{t+1} = H_t - d_t + R_i \tag{14}$$

Where d_t , H_t and R_t are respectively the annual pumping, the height of groundwater and recharges to the aquifer. P_i are the probabilities of a given level of recharge in a future year. This problem is termed the stochastic dynamic programming (SDP) problem and has been the basis of operational and research models for at least thirty years.

5.1 Input Data

Input data includes information on maximum pumping depths, aquifer size, water costs, interest rate, and infiltration coefficients. Two main sources of data are required for this model framework: recharge flows and net marginal revenue.

Recharge flows are drawn from a database covering 32 years of normalized recharge flows into the aquifer. The years included are 1960-1992.

The net marginal revenue is a function of a quadratic derived demand for water and the cost of pumping that depends on the electricity cost and the height that the water has to be pumped. This annual net revenue function for pumping in a given year will be used as the basis of calculating the value (costate) for groundwater as a function of the stock of groundwater in that year. This enables the model to trade off the decision to pump or leave water underground on the basis of the effect on total expected discounted returns.

5.2 Stochastic State Variables

Water recharge is stochastic and depends on seasonal rainfall and runoff. We use a lognormal distribution in this example from which we draw the stochastic recharge probabilities. The lognormal distribution is used as the basis for a z distribution with a mean of 0.052 thousand acre-feet (KAF) and a standard deviation of 1.3443 KAF.

In order to solve the SDP we first discretize the state variable. In this example, the recharge is divided into eight ranges corresponding to the z value ranges of: $-\alpha$, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, α.

The eight inflow quantities with their associated probabilities are:

Midpoint	Probability
0.02996	0.0668
0.034737	0.0919
0.040276	0.1498
0.046699	0.1915
0.054145	0.1915
0.062779	0.1498
0.072789	0.0919
0.084395	0.0668

5.3 SDP Solution

To solve the groundwater SDP we apply the algorithm described in today's notes. The SDP solution in the GAMS code follows the sequential steps, described in detail earlier in the course notes:

a. Define the Chebychev polynomial
b. Define the state-space and map to the [-1,1] interval
c. Iterate to find the fixed point of the Bellman equation while solving for Chebychev coefficients
d. Iterate until convergence tolerance is achieved

The SDP solves after 260 iterations, and converges to a steady state value. The value function that results from the SDP is six degree polynomial with coefficient values defined in the table below.

Polynomial Order	Coefficient
1	7356.97
2	835.87
3	-270.38
4	70.79

5	-16.87
6	2.905

These are the coefficients for the six degree Chebychev polynomial approximation to the value function. We can now run simulations of the optimal solution and evaluate policy response.

5.4 SDP Simulation

Simulation of the optimal solution from an SDP uses the coefficients and functions from the estimation to simulate a future time path. Initial values and the evolution of state variables may be specified from existing data or other sources.

For our example, the value function and the net marginal revenue function are combined to form a net present value function for each year. This is a function of current deliveries and the carry over stock of water. Since the accrual for the next season is not known to the decision-maker at the beginning of the irrigation season, the expected value of the future periods must be used to make an optimal decision under uncertainty. This is calculated by taking the remaining ownership after the deliveries and calculating the ownership in the next year and its value for each of the range of possible inflows. The eight functional values are then weighted by the probability of each inflow level to get the expected function value. Note, that by Jensen's inequality, the expectation of a nonlinear function is not equal to the function of the expected inflow level.

The simulation is initiated using the ownership in 1959, and first optimized for 1960. Each year is optimized separately; the only information is the initial ownership stock of water, the inflow probabilities and quantities, and the delivery value function and corresponding costate function.

After the delivery has been solved by optimization, the actual inflow realized for that year is used to update the ownership to the actual value for the start of the next water year. The process repeats over the horizon of the simulation program.

We can see the intuition behind dynamic problems -- namely, that expectations and future period returns will affect decisions today. In each year the model trades off the current marginal net revenue with the expected value in future years. Future years' expected value is derived as a function of the stock of groundwater remaining for the next year and the probability of different inflow levels.

6 Further Reading

Lars Hein. Case Study: Rangeland Management in the Ferlo, Senegal. In Economics and Ecosystems: Efficiency, Sustainability and Equity in Ecosystems Management. Edwar Elgar. UK.

Wopke van der Werf, Karel Keesman, et al. 2007. Yield-Safe: A parameter-sparse process-based dynamic model for predicting resource capture, growth and production in agroforestry systems. Carbon sequestration and landscape ecology in Western Europe. Issue 4. April 2007.

Day 5: Multi-Market Models

At the end of the day you will be able to:

a. Specify a primal and dual version of an inter-regional trade model
b. Understand the policy significance of inter-regional and inter sector water trade.
c. Understand the theoretical basis of water trade policy
d. Specify and interpret a water trade model
e. Understand the basics of a Multi-Market Model
f. Calibrate urban demands using prior econometric elasticity estimates

1 Introduction

Today we will switch back to a static framework and look at an inter-regional trade model. We will consider inter-regional trade in the context of water transfers.

2 Modeling Inter-regional Trade

There are several different ways of setting up an inter-regional trade problem, but perhaps the most intuitive method is to apply the normal quantity dependent supply / demand concept to *i* exporting regions and *j* importing regions. The regions are linked by trade flows that incur a trading cost of c_{ij} per unit commodity traded from region *i* to region *j*.

2.1.1 Primal Trade Model

We can define quantity-dependent supply and demand functions.

$$(1)\quad pd_j = \phi_j + \delta_j xd_j$$

$$(2)\quad ps_i = \alpha_i + \gamma_i xs_i$$

And we can define the primal inter-regional trade model:

$$(3)\quad \max F(.) = \sum_{j=1}^{J}\left(\phi_j + 0.5\delta_j xd_j\right)xd_j - \sum_{i=1}^{I}\left(\alpha_i + 0.5\gamma_i xs_i\right)xs_i - \sum_i\sum_j c_{ij}x_{ij}$$

subject to

$$(4)\quad xs_i = \sum_j x_{ij}$$

$$(5)\quad xd_j = \sum_i x_{ij}$$

(6) $x_{ij} \geq 0$

Where *xs* and *xd* are quantity supplied and quantity demanded, respectively, and the variable *x* tracks total flows between regions *i* and *j*. α, ϕ, δ and γ are parameters for the linear supply and demand functions for each region. We see that the objective function maximizes the definite integral under the demand and supply function for each region in terms of the post trade quantities demanded and supplied in each region. The costs of trading between regions is deducted to yield the net social benefit of trade shown in the two region diagram. The adding up constraints ensure that the quantities demanded and supplied in each region balance.

2.1.2 Dual Trade Model

The specification of the regional trade problem is an excellent illustration of the efficiency and beauty of dual specifications. The primal specification above solves optimally, but it is a bit more complicated than needed. Since the decision variable is the quantity of product traded between regions, the cost of trading is explicitly defined in the objective function, and the aggregate quantities are generated by the summing up constraints. An alternative to using quantity dependent supplies and demands, is to formulate the dual of the quantity dependent problem that can be termed the price dependent form. The price dependent specification solves the problem with two simple equations that use the standard quantity dependent demand and supply functions, and instead of solving for i*j quantities traded, the price based model solves for the i + j set of equilibrium prices. The two equations are the producer and consumer surplus objective function, and the first order price condition for trade.

We could also define price-dependent demand and supply functions and corresponding unknown parameters,

(7) $xd_j = a_j + s_j pd_j$

(8) $xs_i = b_i + g_i ps_i$

and specify the dual of the inter-regional trade problem (where we <u>minimize</u> with respect to the prices, which are the decision variables now):

(9) $\min F^{dual}(.) = 0.5\sum_{i=1}^{I} \gamma_i \left(b_i + g_i ps_i\right)^2 - 0.5\sum_{j=1}^{J} \delta_j \left(a_j + s_j pd_j\right)^2$

subject to

(10) $pd_j - ps_j - c_{ij} \leq 0$

The quantities traded are generated as the dual values to the (i * j) inter-regional pricing constraints. From the complementary slackness principle we know that if the trade price constraint is slack, that is the supply and transport costs exceed the demand price, then the quantity traded will be zero. The corollary is that the dual value when the trade price constraint is

binding is the quantity traded. Note that all the information from the optimal solution of the primal problem is also obtained from solving the simpler dual problem. The minimized value of the dual problem should also equal the maximized value of the primal problem, at the optimal solution. We will verify this in the GAMS program.

2.2 Example: California Water Trading

We can explore the fundamental concepts of inter-regional trade using the GAMS code for a California water trade model. The program includes 4 regions in California, each with different supply and demand for water.

a. Northern Sacramento Valley
b. Southern San Joaquin Valley
c. Bay Area
d. Greater Los Angeles

The model maximizes the net social benefit of water trade taking into account the cost of transfer between regions. Notice that the GAMS code specifies the dual problem.

2.2.1 California Water Trading Exercises

a. Run the base model as it is set up. Comment briefly on the amounts of water traded. Show that Los Angeles is in spatial trade equilibrium.
b. The base run derived demand for Los Angeles is calculated from the data below. Recalculate the demand for an elasticity of - 0.59 and the same price and quantity. Enter this new demand function into your GAMS program. Comment on the effect on the equilibrium trade solution of the change in elasticity.
    i. Elasticity - 0.89
    ii. Price $172.7 / acre/ft
    iii. Quantity Demanded: 2.9 million acre-feet.
c. Introduce uncertainty in the water supply from the South San Joaquin Valley (Region 2) by assuming that the supply function intercept parameter is distributed normally, with a mean value – 4.63 and variance of 4. Calculate and comment on the effect on the optimal solution from this uncertainty. You will need to parametrically evaluate the model.

3 Multi-Market Models

In this section we will discuss multi-market models which allow us to evaluate linkages and feedback between markets. We will briefly cover Social Accounting Matrices and then a basic multi-market model today.

3.1 Multi-Market Model Overview

Multi-market models are used for partial equilibrium analysis of the impact of changes in prices and quantities in selected markets on other markets. The model may include a system of supply and demand equations for several regions and sectors of the economy. This allows the modeler to analyze impacts of policies in one sector on other sectors of the economy.

Some examples of where multi-market models are useful include:

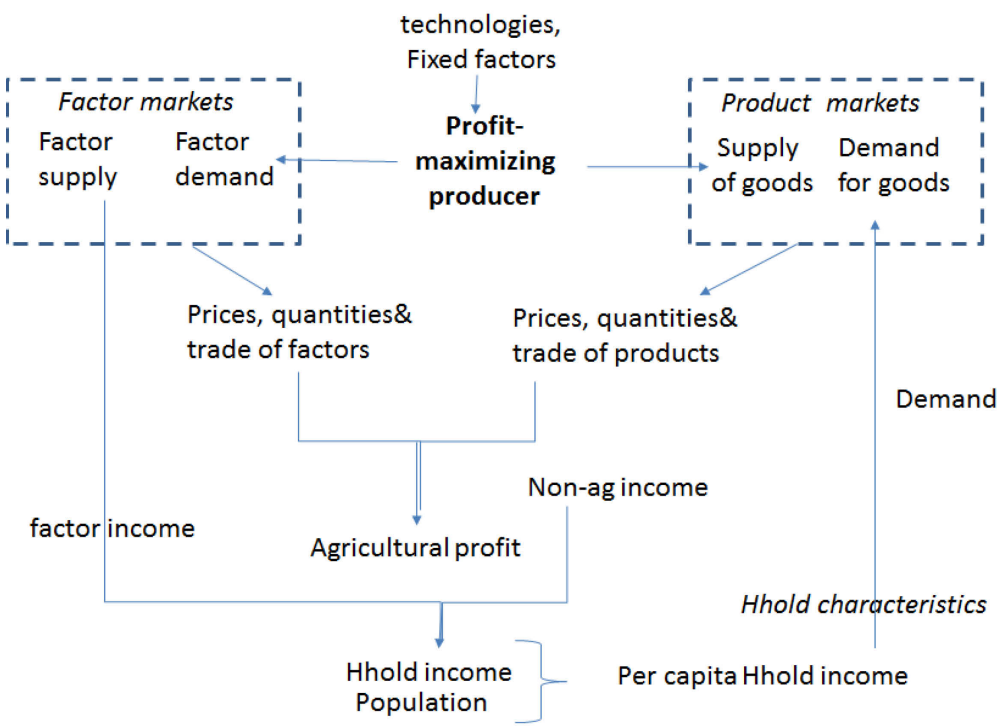
a. Poverty and distributional impact of commodity price policies
b. Distributional impacts of subsidies and taxes
c. Distributional impacts of tariffs and quotas
d. Impacts of changes in prices of imported or exported commodities
e. Evaluation of external shocks and policies on individual sectors of the economy

Typically, there is no standard template for multi-market models. They tend to vary widely in scope and nature, depending on the research question they are designed to address, and the interests of the analyst/researcher who is building it. While some multi-market models ignore factor markets entirely, most try and capture the key factor markets which are relevant – which, for the case of agriculture, would be labor and fertilizer. Some models ignore the flows of revenues between production activities and consumer households, and the interventions of the government – while others try and capture the payment of wages to households, the revenues from taxes that accrue to governments, as well as the transfers that governments make to households. The share of profits that come from production activities and accrue to households can also be captured by the model. Some researchers build their structure very closely on theory – in how the demand and supply of both goods and factors are modeled (and how their functional form is derived from the underlying minimization or maximization problem) – whereas others use fairly ad-hoc specifications in their functions. In general, most multi-market problems maintain the underlying paradigm of agent optimization – although it may not be modeled as explicitly as in the more micro-level problems that we've seen, so far.

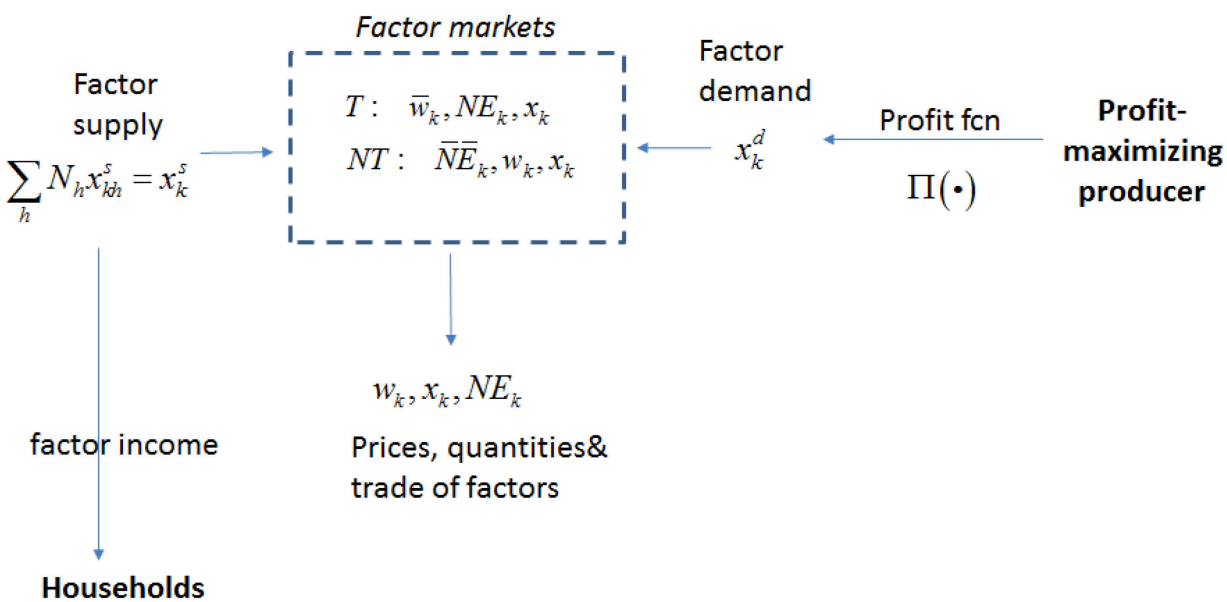
Despite the fairly wide variety that exists in the form and structure of multi-market models, most models contain these basic components:

- Price responsiveness and flexibility
- Multi-agent representation (producers, consumers, exporters, etc)
- Trade (or the non-tradeability) of produced/consumed goods and factors of production.

A simple schematic of how multi-market models are structured is given in the figure below, which is derived from the treatment by DeJanvry and Sadoulet (1995).



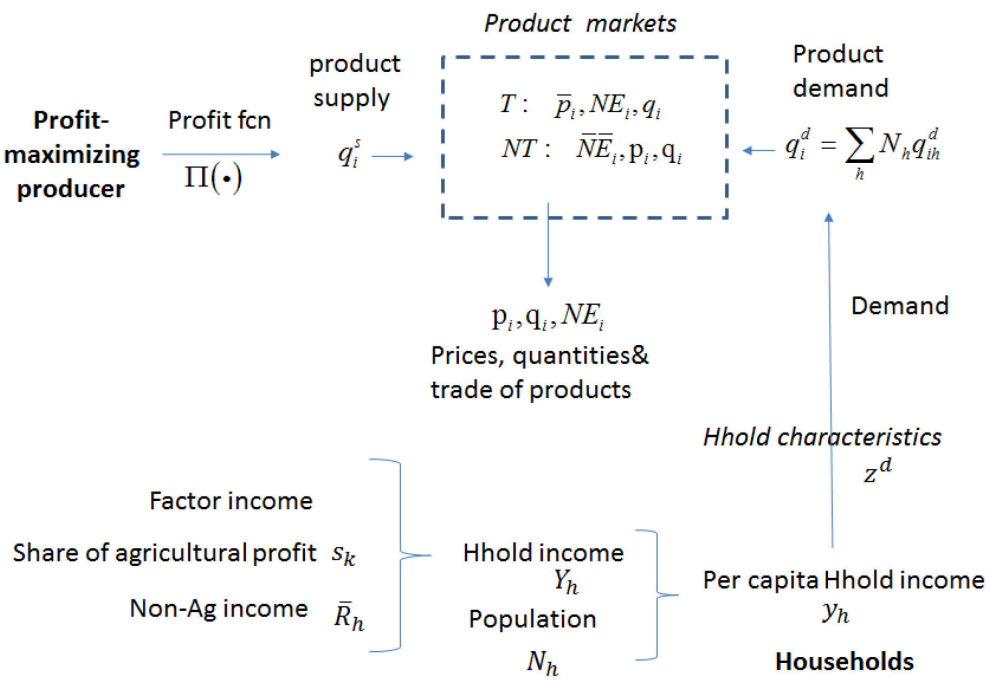
This shows how the household consumption and production sides are connected both through the products and the factors of production. If we were to look more closely at the factor market linkages, we see the details shown below



Where the household is allowed to sell its labor as a factor of production, and receives income from it. The underlying optimization paradigm allows us to derive the factor demands from the

profit function of the producer (which is the closed-form "value function" of profit maximization, as given by micro-economic theory of the firm). This approach is consistent with economic theory and is often the basis upon which econometric estimations of supply response (and factor demand) are made by researchers.

Looking more closely on the production side, we see that the linkages in product markets is given by the schematic below



Where the underlying paradigm of optimization (on the part of all agents in the model) allows us to obtain the product supply from the firms (or farms, in the case of agriculture) and the demands from the consumers (households).

A more detailed treatment of this is given in Chapter 11 of the DeJanvry and Sadoulet book.

3.2 Data Requirements

A multi-market model allows for analysis of the induced substitution effects across selected goods in response to policy and shocks. The modeler must first determine relevant markets where the policy under consideration will have a direct effect. Multi-market models involve a system of equations representing a sub-set of the economy. This includes producers, consumers, and government and the corresponding profit functions, factor and product markets, income to the owners of inputs, and final consumption. Price or quantity equality is imposed for each good in the system of equations to ensure the market clears and the model is closed.

Data required:

a. Disaggregate income and/or consumption data across households
b. Supply and demand functions for all markets in the model (and cross-effects)
c. Specify model closure (domestic and rest-of-world markets)
d. Mathematical specification mapping endogenous variables into the income and consumption of households and other markets

3.3 Example Multi-Market Model: Multi-Good/Region

Now we turn to an example given by Minot (2009) that allows us to consider the case with trade in multiple goods and regions. This model provides a very good basis upon which to understand the role that trade policy instruments can play in determining outcomes in international (and inter-regional) trade, and how they can be modeled. In contrast to the spatial equilibrium model in water that we just considered, this model runs without an explicit objective function or optimization criterion – although it assumes that profit- and utility-maximization are motivating the behavior of the agents. It also maintains certain no-arbitrage conditions that are similar to what we saw in the previous spatial equilibrium example.

Like the spatial equilibrium trade model, this model allows for differences in prices across regions which, in turn, determine the flow of goods between them. The model includes a 7th region which is supposed to represent the rest-of-the-world, thereby allowing us to consider international trade impacts (although international prices are held fixed at exogenous levels). This will allow us to see the effects of exchange rate policies, as well.

We should solidify, first, some key concepts that underlie the price relationships that are modeled in this formulation. The CIF or "Cost-insurance-Freight" price serves as the long-run equilibrium price that maintains an upper-bound on the domestic prices (at the port-of-entry). Why? Because an agent would rather import than pay a higher price for the same good on the domestic market. The FOB or "free-on-board" price is the long-run equilibrium export price that puts a lower-bound on prices seen at the port-of-exit. Why? Because a selling agent would rather export to the international market than sell the same good at a lower price on the domestic market. Both the CIF and FOB prices are the key prices that exist at the border and which are linked to the international price through policy instruments.

Because there are transportation costs of getting goods to the interior of the country, we also have to consider the "export parity" and "import parity" prices on the domestic market. The transportation cost has to be added to the cost of importing a good such that the import parity price (IPP) = CIF + TC (transport costs). This defines the relevant upper-bound on prices that are seen within the interior of the country. Conversely, the transportation costs are subtracted from the export price (for good originating from the interior of the country) – such that the 'export parity price' (EPP) = FOB – TC, and represents the lower bound on prices in the interior of the country. So in the model, we will see that the regional prices will move relative to the IPP and EPP values (and the policies that affect them).

We will also see that trade is determined by the price relationships in the model, as was seen in the case of the simple spatial equilibrium model. We can call the price of the good that exists in the absence of trade as the 'autarky' price (p_a). We would expect that if:

- $p_a > IPP$ then a country would import the good
- $p_a < EPP$ then a country would export the good; and if
- $IPP > p_a > EPP$ then no trade occurs

3.3.1 Making the flow of trade endogenous to the model.

The model also considers the effects of trade policies (taxes and quotas), which change the domestic price relative to the CIF and FOB prices at the border. For instance:

- An import tax raises the domestic price above the CIF value
- An export tax lowers the domestic price below the FOB level
- Quotas on imports will act to restrict the domestic supply levels, and raise prices – thereby acting like an implicit tax on imports
- Quotas on exports will raise the domestic supply (relative to the case without any quota/limit) and thereby lower prices, which is like an implicit tax on exported goods.

The model will have both explicit and implicit export and import taxes – where the taxes determined by policies are represented by explicit fixed parameters, whereas the implicit taxes come from the constraints defining the quotas.

The key equations of the model are given below:

$$Supply + \sum_{other\,\mathrm{Regs}} (inflows) - \sum_{other\,\mathrm{Regs}} (outflows) + iMport = Demand + eXprt$$

Domestic market balance

$$P_{reg} + TC + \mathrm{Im}\,Tax_{\mathrm{export}} \geq P_E$$

Export price relationships

$$P_M + TC + \mathrm{Im}\,Tax_{im\,\mathrm{port}} \geq P_{reg}$$

Import price relationships

$$Demand = f_D(P, \mathrm{y})$$

Domestic demand

$$Supply = f_S(P)$$

Domestic supply

$$P_{regionR} + TC_{R,RR} \geq P_{regionRR}$$

Domestic trade price relationships

$$Quota(\mathrm{X}) \geq \sum_{RR} X_{R,RR}$$

Quota on exports

$$Quota(I) \geq \sum_{RR} M_{R,RR}$$

Quota on imports

$$P_x = NER \times P_{world} \times (1-t)$$

Effect of export tax on effective FOB price

$$P_M = NER \times P_{world} \times (1+t)$$

Effect of import tax on effective CIF price

Endogenous variables:

$$X, M, P_{reg}, \mathrm{inflows}, \mathrm{outflows}, \mathrm{ImTax}_{\mathrm{exp}orts}, \mathrm{ImTax}_{im\mathrm{p}orts}$$

$$Demand, Supply$$

Fixed parameters:

$$P_M, P_X, P_{world}, Quota(I), Quota(X), \mathrm{TC}, \mathrm{t}, \mathrm{NER}$$

The GAMS code that implements this model will be run, during the course, so that we can do various experiments to explore its properties and illustrate the effects of the policy instruments.

4 Further Reading

Minot N. 2009. Using GAMS for Agricultural Policy Analysis. International Food Policy Research Institute.(manuscript available at: http://www.ifpri.org/sites/default/files/publications/gams_trainingmanual.pdf).

Sadoulet E, A de Janvry. 1995. Quantitative Development Policy Analysis. Johns Hopkins University Press.(manuscript available at: are.berkeley.edu/~sadoulet).