

Association for Information Systems

**AIS Electronic Library (AISeL)**

---

ICEB 2001 Proceedings

International Conference on Electronic Business  
(ICEB)

---

Winter 12-19-2001

## **Application of Quantitative Software in Economics and Decision Science**

Mohammad R. Safarzadeh

Hamid Falatoon

Paul Sarmas

Follow this and additional works at: <https://aisel.aisnet.org/iceb2001>

---

This material is brought to you by the International Conference on Electronic Business (ICEB) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ICEB 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

## APPLICATION OF QUANTITATIVE SOFTWARE IN ECONOMICS AND DECISION SCIENCE

Mohammad R. Safarzadeh  
Professor  
Economics Department  
California State Polytechnic University, Pomona

Hamid Falatoon  
Professor  
Management and Business Department  
University of Redlands

Paul Sarmas  
Professor  
Finance, Real Estate, and Law Department  
California State Polytechnic University, Pomona

### ABSTRACT

Mathematics is often used to facilitate a clear and organized presentation of economic theories and problems. It provides a simplified approach for setting up models, making explicit assumptions about the models, finding optimal solutions, and extending the results by varying the parameters and assumptions of the models. Many economists would agree that a large gain in clarity and economy of effort can be achieved by incorporating mathematics into economic teaching. What limits the use of mathematics in teaching economics is inadequate preparation of students in quantitative methods. In addition, some mathematical manipulations and calculations turn out to be time consuming and tedious for class presentations. This paper argues that both of these problems can be overcome by using modern mathematical software in teaching economics.

### I- Introduction

As a tool, mathematics can be used to facilitate a clear and organized presentation and test of economic problems. It provides a basis for setting up a model, making explicit assumptions about the model, finding specific optimal solutions, and extending the results by varying the parameters and assumptions of the model. A large gain in clarity and economy of effort can be achieved by incorporating mathematics into economic analysis.

What limits the use of mathematics in teaching undergraduate courses in economics is students' preparation in mathematics. It is a known fact that many students entering higher educational institutions do not have enough preparation in mathematics to allow them to manipulate mathematical symbols, to do numerical calculations and to graph the mathematical relationships. These techniques are the basic tools of economic analysis and the need for them is of even greater importance in the field of applied economic. In many colleges, the lack of mathematical preparation has resulted in canceling quantitative parts of economic curriculum, lowering the course standards, and omitting some of the advanced topics. Even if students had enough preparation in mathematics, presenting certain topics in the classroom using the traditional methods of on-the-board manipulation is often very time consuming and tedious.

Software utilization in the classroom has the potential to radically transform teaching style from a static lecture walk and talk format to a technologically dynamic environment with an interactive and collaborative learning process. Students not only have the benefit of learning from lectures but also have access to expert online resources implemented in the software that makes the possibility of self study even more likely. The use of quantitative software creates a learning community in which the differentials in quantitative-skill levels become a secondary to conceptual understanding of course materials and application of the knowledge to real world problems.

Since the introduction of mathematical software programs, *Mathematica*, *Maple*, *MathLab*, and others have become an increasingly standard computational environment for scientists, engineers, financial analysts and many others. However, economists have rarely used these software programs in teaching. The use of these programs in teaching can help to remove students handicap in mathematical analysis and allow them to concentrate more on the economic concepts, theories, and the structure of economic optimization rather than worrying about details

of numerical calculations and symbolic manipulations. Using these powerful and user friendly software programs which can handle symbolic, numeric, and stochastic economic models, students can experiment with the dynamics of economic theories. The two-dimensional and three-dimensional graphic features of these programs can help students with visualizing the relationship among different economic variables and can be used in presenting graphical solutions of economic problems.

Today, it is critical for students of business and economics to be fully familiar with mathematical concepts and to be equipped with major mathematical software programs that can help them in modeling a problem, finding the solution, and analyzing it by varying the assumptions and parameters of the problem. The application of quantitative software in teaching economics excels economic students in learning and practicing economics and motivate them to become competent in economic theory and decision process.

*Mathematica* has been utilized in formulating and analyzing several advanced economic and financial models (Varian, 1992 and 1996, ). However, there are no literature that we know of which has addressed the pedagogical benefits of using quantitative software in teaching and learning quantitative subjects in economics and decision science. This paper uses several features of *Mathematica*, including its power in performing numerical calculations, graphic presentation, and symbolic solutions of equations to present cases in which the use of quantitative software can be of enormous benefit in teaching and learning environment. The ease with which the numerical and graphic solutions can be done by a quantitative software and the similarities between the mathematical formulations and the software modeling is an argument for using quantitative software in teaching economics.

## II- Numerical Calculations: Present Value Problem

One of the topics that easily yields itself to numerical calculation power of quantitative software is the present value concept. Suppose a stream of \$R is accrued for the next n yeas. The present value of the cash flows at a discount rate of r can be formulated in *Mathematica* as follows:

```
Clear [n, t, r, PV]
r = ?; n = ?; R = ?;
```

$$PV = \sum_{t=1}^n \frac{R}{(1+r)^t}$$

The present value formulation in *Mathematica* is quite the same as you may find in any standard text book. As well, a decision maker or instructor can set the values of the parameters or change the values to demonstrate the relationship among the three parameters of the problem. For example, with a numerical values of r = 6%, n = 10 years, and R = \$100, the PV = \$736.01. If the stream of the values are accrued over the lifetime (consol), n = ∞, and the present value of the consol equals to \$1666.67. The *Mathematica* solution is done as follows:

```
Clear [n, t, r, PV]
r = .06; n = Infinity; R=100;
```

$$PV = \sum_{t=1}^n \frac{R}{(1+r)^t}$$

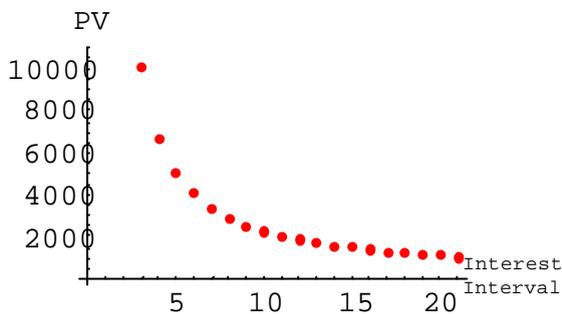
```
{{ $1666.67 }}
```

The graph of the present value of stream of \$100 annual payments forever as the discount rate changing from zero to 10%, with an increments of .05% can be plotted using the *Mathematica* command "ListPlot" as:

```
Clear [n, t, r, PV]
R = 100; n = Infinity;
```

$$PV = \sum_{t=1}^n \frac{R}{(1+r)^t} ;$$

```
pvalue = N[Table [PV, {r, 0.1, .005} ] ] ;
ListPlot [pvalue]
```



**Present Value and Growth**

The present value problem above, can be extended to include growth of the future earnings. Suppose the stream of future cash flows is growing at an annual rate of g per cent. The present value calculation with the revenue growth assumption can be formulated as follows:

Clear [R, r, g, t, n]  
 R = ?; g = ?; r = ?; n = ?;

$$GPV = \sum_{t=1}^n \frac{R(1+g)^t}{(1+r)^t}$$

With a hypothetical numerical assumption of initial revenue R = \$100, n = 10 years, g = 10%, and r = 6%, the present value is \$1232.91.

Clear [R, r, g, t, n]  
 R = 100; g = .10; r = .06; n = 10;

$$GPV = \sum_{t=1}^n \frac{R(1+g)^t}{(1+r)^t}$$

{{1232.91}}

**Internal Rate of Return**

The "Solve" command of the *Mathematica* is a powerful tool in finding internal rate of return (IRR) for any investment project and its future stream of cash flows. The tool can be easily applied in project evaluation, profitability indexing, and terminal value calculations. For example, to find the internal rate of return for an initial investment of CFo = \$24000, with the

stream of the cash flows (NCF) for the next five years; NCF1 = \$5000, NCF2 = \$7000, NCF3 = \$8500, NCF4 = \$6000, and NCF5 = \$3500, the Internal rate of return is calculated as follows,

Clear [NCF1, NCF2, NCF 3, NCF4, NCF5, NCFo, r, IRR]  
 NCF1= 5000; NCF2= 7000; NCF3= 8500;NCF4=6000; NCF5=3500; CFo=2400;

$$IRR = N[\text{Solve}[\frac{NCF1}{(1+r)} + \dots + \frac{NCF5}{(1+r)^5} - CFo = 0, r]]$$

{{r -> 0.08279}}

The formulation in *Mathematica* follows exactly the same format of the IRR formulation in any standard textbook.

If the stream of cash flows for a project is fixed with a constant growth rate of g, the IRR is calculated as follows:

Clear [NCF, CFo, g, r, n]  
 CFo = 24000; NCD = 800; g = .1; n = 20;

$$\text{Solve}[\sum_{t=1}^n \frac{NCF(1+g)^t}{(1+r)^t} - CFo = 0, r];$$

{{r -> 0.0605}}

where, the internal rate of return is 6.05%. By changing the values of the parameters of the equation, a decision maker or an instructor can evaluate the resulting values with minimum efforts. For example, the problem may be solved for n, the number of years needed for the project to achieve break even.

Clear [NCF, CFo, g, r, n]  
 CFo = 24000; NCD = 800; g = .1; r = .0605;

$$\text{Solve}[\sum_{t=1}^n \frac{NCF(1+g)^t}{(1+r)^t} - CFo = 0, n];$$

{{n -> 19.9908}}

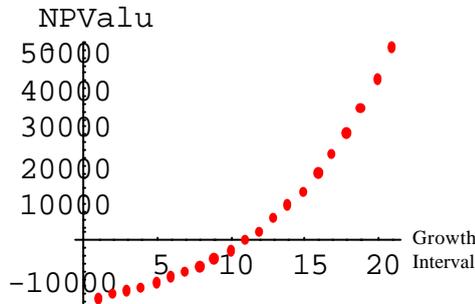
The "ListPlot" command of *Mathematica* can be used to visually present the effect of changes in

growth rate, internal rate of return, initial investment, or any other parameters of the problem on the outcome of the project. For example, a change in annual revenue growth from 0% to 20%, with annual increments of 1% will result in the following net present value plot.

```
Clear [NCF, CFo, g, r, NPV, n]
CFo = 24000; NCF = 800; r = .0605; n = 20
```

$$NPV = \sum_{t=1}^n \frac{NCF(1 + g)^t}{(1 + r)^t} - CFo;$$

```
NPValue=N[Table[NPV, {g, 0,.2,.01}]];
ListPlot[NPValue]
```



### III. Using Graphic Property of Software in Linear Programming:

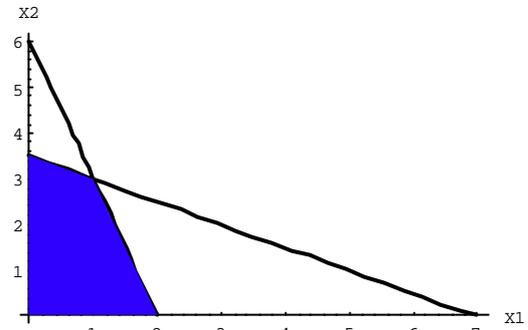
The graphic solution of the linear programming problems is a useful method to illustrate the way in which the Simplex method arrives at the optimal solution. The iterations to achieve optimal final solution starts from the degenerate corner solution at the origin and progresses to optimal solution in several iterations, with each iteration improving the solution over the previous iteration. To illustrate the use of a quantitative software in teaching the linear programming problem, let's start with a standard linear programming problem given as:

Maximize:  $Z = 1.5x_1 + x_2$   
 Subject to:  $x_1 + 2x_2 \leq 7$   
 $3x_1 + x_2 \leq 6$   
 $x_1, x_2 \geq 0$

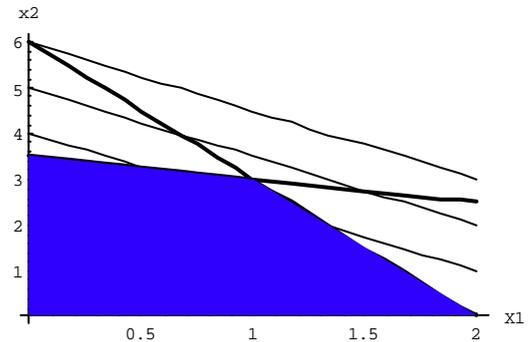
The first step in solving the problem is to find the **constraint region**. The constraint region is

the set of points that simultaneously satisfies the constraints. To find the constraint region, graph both constraints using the *Mathematica* command "Plot" as follows:

```
Clear[x1,x2, equ1, cons1]
equ1=Solve[x1+2x2==7,
x2];eq1[x1_]:=x2/.equ1[[1]];
cons1=Plot[eq1[x1], {x1, 0, 7}];
equ2=Solve[3x1+x2==6,
x2];eq2[x1_]:=x2/.equ2[[1]];
cons2=Plot[eq2[x1], {x1, 0, 2}];
cons = Show[cons1, cons2]
region= Show[Graphics[ {Hue[.7], Polygon[{{0,
0}, {0, 3.5}, {1,3}, {2,0}}]}]]
Show[cons , region]
```



The Simplex method provides us with a prove that the optimum solution of the linear programming problems always lies on one of the corner points of the polygon or on one of the sides of the polygon, if there are multiple solutions. The optimum solution, however, depends on the slope of the objective function and the point at which the objective function is tangent to the constraint region. The constraint region and a group of objective lines are graphed below



as the objective line is moved from the north east corner of the quadrant toward the constraint region, the first point where the objective line touches the constraint region (the point of tangency) provides us with the optimum solution

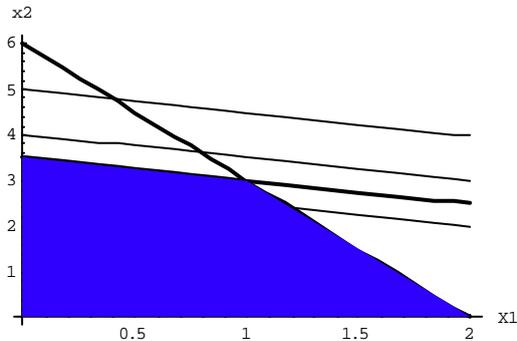
to the problem. In this case, the solution is (1, 3), which is the intersection of the two constraints, and the maximum value of Z is Z = 11.

The optimum solution can be tested for using the *Mathematica* command "ConstrainedMax" as follows,

```
Clear[x1, x2]
ConstrainedMax[2x1+ 3x2,{x1 + 2x2 <= 7,3x1 + x2 <= 6},{x1,x2}]
{11,{x1->1,x2->3}}
```

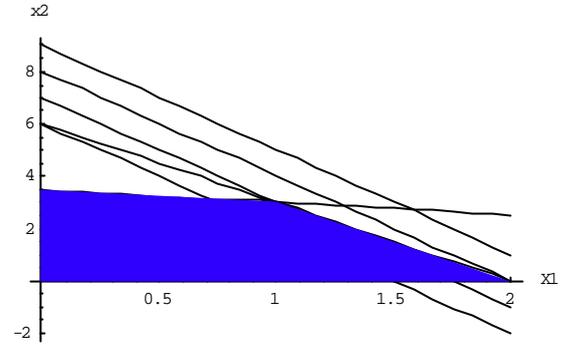
It may happen that the slope of the objective function is the same as the slope of one of the constraints. In that case, the linear programming problem will have **multiple solutions**. Any point on the constraint function which also lies on the constraint region will be a solution to the problem. For example, if the objective function of the preceding example were  $Z = x1 + 2x2$ , which has the same slope as the first constraint, the point (0, 3.5), or (1,3), or any point on the line joining these two points is a solution to the problem. All of these points will maximize Z with a value of Z = 7. The following figure shows the case of multiple solutions.

```
Show[Plot[{eq1[x1], eq2[x1]}, {x1, 0, 2}],
Table[Plot[-.5x1+i, {x1, 0, 2}, DisplayFunction->Identity], {i, 3, 5}],Graphics[{{Hue[.7],
Polygon[{{0, 0}, {0, 3.5}, {1,3}, {2,0}}]}]]
```



If the objective line intersects the constraint region in one of the vertices lying on the axes (point (2, 0) or (0, 3.5)), the firm will produce positive amounts of one good and none of the other. Such a solution is called a **degenerate solution**. For example, if the objective function was  $Z = 4x1 + x2$ , the solution would be  $x1 = 2$  and  $x2 = 0$ . If the objective function was  $Z = x1 + 3x2$ , the solution would be  $x1 = 0$  and  $x2 = 3.5$ . The following figure shows the degenerate solution (2, 0).

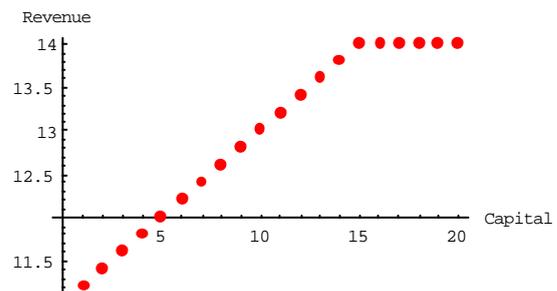
```
Show[Plot[{eq1[x1], eq2[x1]}, {x1, 0, 2}],
Table[Plot[-4x1+i, {x1, 0, 2}, DisplayFunction->Identity], {i, 6, 9}],Graphics[{{Hue[.7],
Polygon[{{0, 0}, {0, 3.5}, {1,3}, {2,0}}]}]]
```



### Sensitivity Analysis

For the maximizing problem above, let's relax the assumption of the fixed inputs and assume that the second input (capital) can be increased. Given that the first input (labor) is fixed, the question is what will be the effect of increase in capital on total revenue, on outputs (X1 & X2), and what is the limit of the increase in capital. Formulating the increments in capital to the optimizing problem gives,

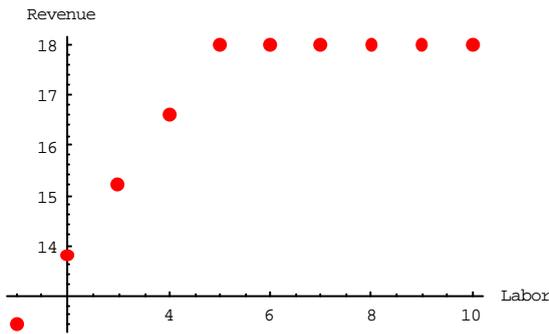
```
Clear[z,u, x1, x2]
fu=Table[ConstrainedMax[ 2x1+3x2,{x1 + 2x2 <= 7,3x1 + x2 <= 6+i}, {x1, x2}], {i, 20}];
u=N[Table[List[i, fu[[i, 1]]], {i, 20}]]
```



The solution and the graph above show that as capital input increases total revenue will increase. With labor input fixed, maximum total revenue is achieved when capital input is 22 units, where  $X1=7$ ,  $X2 = 0$ , and the total revenue is \$14.

Now let the first input (labor) be variable and the second input (capital) be fixed. Formulating the problem into the constrained optimizing problem gives:

```
Clear[z,u, x1, x2]
fu=Table[ConstrainedMax[ 2x1+3x2,{x1 + 2x2
<= 7+i,3x1 + x2 <= 6}, {x1, x2}], {i, 10}];
u=N[Table[List[i, fu[[i, 1]]], {i, 10}]];
ListPlot[u]
```



The solution and the graph above show that as labor input increases total revenue will increase. With capital input fixed, maximum total revenue is achieved when labor input is 12 units, where  $X_1=0$ ,  $X_2=6$ , and the total revenue is \$18.

**IV. Constrained Optimizing Problem**

Solving even a simple numerical example of a constrained optimization problem in the classroom is a time consuming task that every faculty and student would like to avoid. The similarity between *Mathematica* and mathematical formulation and the ease with which the problems can be solved with quantitative software provide faculty and students with a powerful tool in the area of solving optimization, dynamic problems, and simulations. Consider a constrained maximization problem, where the constraint is an equality.

Maximize:  $U = U(X, Y)$

Subject to:  $P_x X + P_y Y = M$

Where the first equation is the objective function in which  $U(X, Y)$  is the utility function, assumed to be differentiable, and  $P_x X + P_y Y = M$  is the budget constraint. The problem is to maximize

the utility function subject to the budget constraint.

The solution of this problem follows the standard constrained optimization techniques which uses Lagrangian function (L), which is formed by augmenting the objective function and the constraint together with the Lagrange multiplier ( $\ell$ ).

$$L = U(X, Y) + \ell (M - P_x X - P_y Y)$$

The mathematical solution to this problem can be solved by optimizing the Lagrangian function  $L(X, Y, \ell) = U(X, Y) + \ell (M - P_x X - P_y Y)$ . This function may be maximized with respect to the three unknown variables X, Y, and  $\ell$ . The first order condition is:

$$\begin{aligned} L_x &= U_x - \ell P_x = 0 \\ L_y &= U_y - \ell P_y = 0 \\ L_\ell &= M - P_x X - P_y Y = 0 \end{aligned}$$

A point  $P(X^*, Y^*, \ell^*)$  that satisfies the three equations of the first-order condition is the extremum.

The second order condition for  $L(X, Y, \ell)$  to have a maximum at P is that  $d^2L$  to be negative definite at the extremum. To find  $d^2L$ , find total differential of  $dL$  with respect to three variables. Expressed in matrix form, the second-order condition gives,

$$(dX \ dY \ d\ell) \begin{pmatrix} U_{xx} & U_{xy} & -P_x \\ U_{yx} & U_{yy} & -P_y \\ -P_x & -P_y & 0 \end{pmatrix} \begin{pmatrix} dX \\ dY \\ d\ell \end{pmatrix}$$

The quadratic form of the second-order condition is said to be negative definite if the bordered principal of the bordered Hessian,  $H =$

$$\begin{pmatrix} U_{xx} & U_{xy} & -P_x \\ U_{yx} & U_{yy} & -P_y \\ -P_x & -P_y & 0 \end{pmatrix} \text{ alternate in sign,}$$

beginning with the first bordered principal minor negative. That is, the second-order condition for maximum utility at the point  $P^*$  requires that

$$\text{Det} \begin{pmatrix} U_{xx} & -P_x \\ -P_x & 0 \end{pmatrix} < 0 \quad \text{and}$$

$$\text{Det} \begin{pmatrix} U_{xx} & U_{xy} & -P_x \\ U_{yx} & U_{yy} & -p_y \\ -P_x & -p_y & 0 \end{pmatrix} > 0.$$

Now let's formulate and solve a numerical example of the same problem using *Mathematica*. Suppose a consumer has a utility function expressed as  $U(x, y) = \ln xy$ , where  $x$  and  $y$  are quantities of two goods  $X$  and  $Y$  consumed by the consumer. The market price of  $X$  and  $Y$  are \$2 and \$3, respectively. The consumer has a budget of \$120 to spend on the two goods. To solve the utility maximizing problem for the consumer and find the optimum quantities of  $x$  and  $y$ , form the Lagrangian function in *Mathematica* format as:

```
Clear[L, U, x, y, l]
L = Log[x y] + l (120 - 2x - 3y);
sol = Solve[{D[L, x]==0, D[L, y]==0, D[L, l]==0}, {x, y, l}]
Clear[L, U, x, y, l]
L = Log[x y] + l (120 - 2x - 3y);
sol = Solve[{D[L, x]==0, D[L, y]==0, D[L, l]==0}, {x, y, l}]
```

The first line above clears the values of the variables from the computer memory. The second line formulates Lagrangian function. The third line derives the first order condition by finding the first derivatives of  $L$  and by setting them zero. The resulting equations then are solved simultaneously and the solution set is assigned a name, here "sol". To find the second-order conditions, formulate the Hessian matrix in *Mathematica* format at evaluate it at "sol", the solution set of the first order condition.

```
{{l -> 1/60, x -> 30, y -> 20}}
MatrixForm[
  {
    {D[D[L, x], x] D[D[L, x], y] D[D[L, x], l]},
    {D[D[L, y], x] D[D[L, y], y] D[D[L, y], l]},
    {D[D[L, l], x] D[D[L, l], y] D[D[L, l], l]}
  }
]
```

$$\left\{ \left\{ \begin{pmatrix} -\frac{1}{900} & 0 & -2 \\ 0 & -\frac{1}{400} & -3 \\ -2 & -3 & 0 \end{pmatrix} \right\} \right\}$$

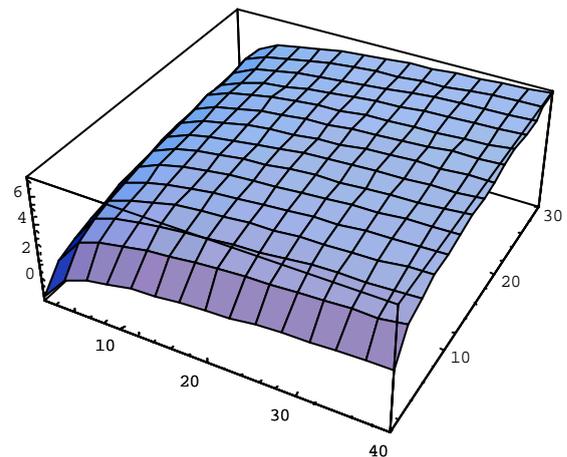
The determinant of the bordered principal minors are formulated as below and are evaluated at "sol",

```
N[Det[
  {
    {D[D[L, x], x] D[D[L, x], l]},
    {D[D[L, l], x] D[D[L, l], l]}
  }
]]/.{sol}
{{-4.}}
N[Det[
  {
    {D[D[L, x], x] D[D[L, x], y] D[D[L, x], l]},
    {D[D[L, y], x] D[D[L, y], y] D[D[L, y], l]},
    {D[D[L, l], x] D[D[L, l], y] D[D[L, l], l]}
  }
]]/.{sol}
```

{0.02} Since the determinants of the bordered principal minors alternate in sign beginning with the first one negative, the Hessian is negative definite and the extremum is a maximum.

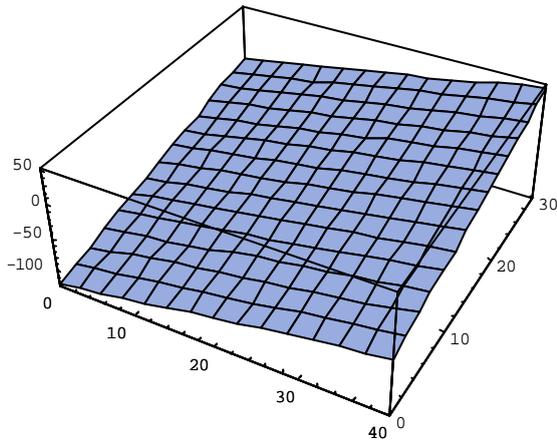
To see the solution point graphically in three dimensions, plot the objective function and the constraint together

```
U = Log[x y];
plo1 = Plot3D[U, {x, .5, 40}, {y, .5, 30}]
```



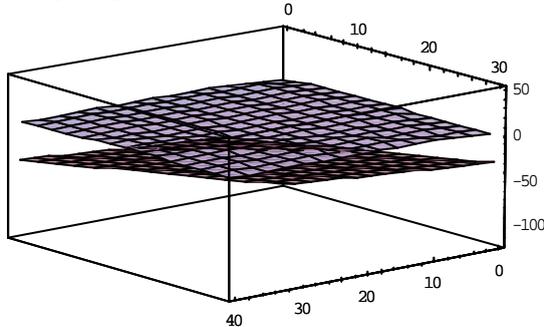
The plot of the constraint surface is derived as:

```
plo2 = Plot3D[2x + 3y - 120, {x, 0, 40}, {y, 0, 30}]
```



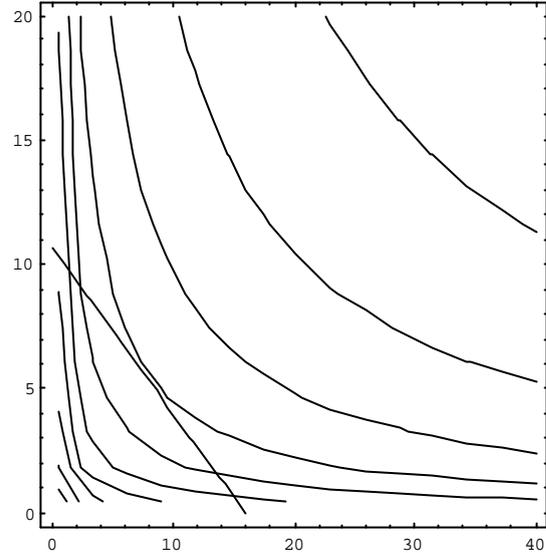
The tangency of the utility function and the budget constraint is shown in three dimensions using the command

```
Show[plol, plo2, ViewPoint->{33, 41, 10.6}]
```



To see the solution point graphically in x-y space, plot the indifference curves and the budget constraint through the contour plot:

```
Clear[u, x, y]
con1 = ContourPlot[U, {x, .5, 40}, {y, 5, 20},
ContourShading->False]
con2 = ContourPlot[2x+3y-32, {x, 0, 40}, {y, 0,
20}, Contours->{0}, ContourShading->False]
Show[con1, con2]
```



The Lagrangian multiplier technique may be generalized to cases in which a function is optimized subject to more than one constraint, provided that the constraints are consistent and the number of constraints is less than the number of variables. The *Mathematica* solution to the m-constraint case follows the same formulation as in this section, with derivatives extended to the new variables introduced to the model.

## V- CONCLUDING REMARKS

Advances in semi-conductor and information technology have breathed a new life into the abstract formulas, particularly in decision theory, by making it possible for us to visualize the concepts in sequential, simplifying, and accessible graphical presentations with much flexibility in a timely fashion in the classroom environment. Given the increasing importance of quantitative decision theory in the coming decades, information technology can assist both students and instructors in spending more of their efforts on the analytical issues rather than on routine computational tasks. It is time for instructors to take full advantage of the technology in the classroom instructions and concentrate more on the concepts and theories than on mathematical computations.

## VI- END NOTES

1- The benefits from application of *Maple* in teaching economics has been discussed in a recent article published in the Journal of Economic Education (Boyd, 1998).

2- If the number of the constraints is equal to the number of the variables, locating the extremum will become a trivial matter of finding the intersection point of the constraints.

3-You may have noticed that there are two differences between pure income effect and income effect of price change. First, they have different signs; second, income effect of price change is multiplied by  $x$ . Hence, from the pure income effect, we can get information about the sign of the income effect of price change.

## VII. REFERENCES

Boyd, David W., "On the Use of Symbolic Computation in Undergraduate Microeconomics Instruction," Journal of Economic Education, Summer, 1998.

Leuthold, Jane H., "Building a Homepage for Your Economics Class," Journal of Economic Education, Summer, 1998.

Shaur, William, *Modeling Financial Derivatives with Mathematica*. Nomura Research Institute, London, Europe, 1999.

Varian, Hal R. *Economic and Financial Modeling with Mathematica*, Telos, The Electronic Library of Science, 1993.

Varian, Hal R. *Computational Economics and Finance; Modeling and Analysis with Mathematica*, Telos, The Electronic Library of Science, 1996.

Wolfram, Stephen. *The Mathematica Book*, 3rd ed. Wolfram Media/Cambridge University Press, 1996.