#Input Side Economics: Complete Worked out file for Profit Maximization, Cost Minimization and Output Maximization** (**Expenditure Demand Function**)* **and** ***comprative statics using Quadratic **and** ***Cobb Douglas Production Function*** **in** ***Perfectly Competitive Market Maple Code. Author*** **:** ***Bijesh Mishra.#***

restart;
Digits := (3); *#Limit upto three digits after decimal.*

$$Digits := 3 \tag{1}$$

$r1 := 8$; $r2 := 10$; $p := 10$; $b := 0$; $Co := 936$; $yo := 385$;
 # labor, capital, output price, fixed cost, investment, output demanded.

$$r1 := 8$$
$$r2 := 10$$
$$p := 10$$
$$b := 0$$
$$Co := 936$$
$$yo := 385 \tag{2}$$

 # Cobb Douglas Production Function:

$Z := z$; $m := m$; $n := n$;
 # change values here to change the Cobb Douglas function equation given by eq. 3. and change eq. 6 to "cobb" to run optimization using cobb douglas production function with two input and onw output.

$$Z := z$$
$$m := m$$
$$n := n \tag{3}$$

$cobb := Z \cdot x1^{m} x2^{n}$; *#Cobb Douglas Production Function.*

$$cobb := z\, x1^{m}\, x2^{n} \tag{4}$$

 # Quadratic Production function (two input one output):

$b1 := 6$; $b2 := 9$; $c1 := -0.2$; $c2 := -0.3$; $d1 := 0.4$; $a := 0$;
 # change values here to change the quadratic equation and set up y to "quad" to run optimization using quadratic production function with two input and one output. Same procedure works for Cobb Douglas production function which can be done by simply inserting Cobb douglas production function instead of quadratic function.

$$b1 := 6$$
$$b2 := 9$$
$$c1 := -0.2$$
$$c2 := -0.3$$
$$d1 := 0.4$$
$$a := 0 \tag{5}$$

$quad := b1 \cdot x1 + b2 \cdot x2 + c1 \cdot x1^2 + c2 \cdot x2^2 + d1 \cdot x1 \cdot x2 + a$;

$$quad := -0.2\, x1^2 + 0.4\, x1\, x2 - 0.3\, x2^2 + 6\, x1 + 9\, x2 \tag{6}$$

 # Change function to quadratic (quad) or Cobb Douglas (cobb) production function with one output and two inputs. This process works for both types of function.

$y := quad;$

$$y := -0.2\,x1^2 + 0.4\,x1\,x2 - 0.3\,x2^2 + 6\,x1 + 9\,x2 \tag{7}$$

$APP1 := simplify\left(\frac{y}{x1}\right); APP2 := simplify\left(\frac{y}{x2}\right);\ AVP1 := simplify(APP1 \cdot p)\ ; AVP2 := simplify(APP2 \cdot \mathbf{p});$

 # APP1 & APP2 are average physical productivity. AVP1 & AVP2 Average value productivity which is obtained by multiplying price of output with respective APPs.

$$APP1 := \frac{-0.2\,x1^2 + (0.4\,x2 + 6.)\,x1 - 0.3\,x2^2 + 9.\,x2}{x1}$$

$$APP2 := \frac{-0.3\,x2^2 + (0.4\,x1 + 9.)\,x2 - 0.2\,x1^2 + 6.\,x1}{x2}$$

$$AVP1 := \frac{-2.\,x1^2 + (4.\,x2 + 60.)\,x1 - 3.\,x2^2 + 90.\,x2}{x1}$$

$$AVP2 := \frac{-3.\,x2^2 + (4.\,x1 + 90.)\,x2 - 2.\,x1^2 + 60.\,x1}{x2} \tag{8}$$

$f1 := \frac{\partial}{\partial x1}(y);\quad f2 := \frac{\partial}{\partial x2}(y);\ MVP1 := p \cdot f1; MVP2 := p \cdot f2;\ MFC1 := MVP1; MFC2 := MVP2;$ #***MPP1, MPP2, MVP1, MVP2, MVP1, MVP2***

$$f1 := -0.4\,x1 + 0.4\,x2 + 6$$

$$f2 := 0.4\,x1 - 0.6\,x2 + 9$$

$$MVP1 := -4.0\,x1 + 4.0\,x2 + 60$$

$$MVP2 := 4.0\,x1 - 6.0\,x2 + 90$$

$$MFC1 := -4.0\,x1 + 4.0\,x2 + 60$$

$$MFC2 := 4.0\,x1 - 6.0\,x2 + 90 \tag{9}$$

$f11 := \frac{\partial}{\partial x1}(f1);\ f22 := \frac{\partial}{\partial x2}(f2);\ f12 := \frac{\partial^2}{\partial x1\,\partial x2}(y);\ f21 := \frac{\partial^2}{\partial x2\,\partial x1}(y);$ #***SOC of f1, #SOC of f2.***

 # f12 & f21 are Factor Inerdependence (competitive, independent or complementary).

$$f11 := -0.4$$

$$f22 := -0.6$$

$$f12 := 0.4$$

$$f21 := 0.4 \tag{10}$$

$MRTS := simplify\left(\left(\frac{f1}{f2}\right)\right);$ #*Marginal Rate of Technical Subsitution* $\cdot\,(MRTS_{21})$

$$MRTS := \frac{-0.4\,x1 + 0.4\,x2 + 6.}{0.4\,x1 - 0.6\,x2 + 9.} \tag{11}$$

$\frac{r1}{r2} = \frac{f1}{f2}$ ***# Marginal Revenue = Marginal Rate of Technical Subsitution (MR = MRTS₂₁)***

$$\frac{4}{5} = \frac{-0.4\,x1 + 0.4\,x2 + 6}{0.4\,x1 - 0.6\,x2 + 9} \tag{12}$$

$SOC := simplify(f2 \cdot f2 \cdot f11 - 2 \cdot f1 \cdot f2 \cdot f12 + f1 \cdot f1 \cdot f22)$; **#Second order condition.**

$$SOC := -0.032\, x1^2 + (0.064\, x2 + 0.96)\, x1 - 0.048\, x2^2 + 1.44\, x2 - 97.2 \tag{13}$$

$Curvature := simplify\left(\left(\frac{1}{f2^3}\right) \cdot SOC\right)$; **# Curvature.**

 # This curvature is derived from production function. Gives change in slope of the isoquant and used to determine convexity of the isoqunt. Also, if you know the equation of isoquant, then curvature is the second derivative of the equation for isoquant.

$$Curvature := \frac{-0.032\, x1^2 + (0.064\, x2 + 0.96)\, x1 - 0.048\, x2^2 + 1.44\, x2 - 97.2}{(0.4\, x1 - 0.6\, x2 + 9)^3} \tag{14}$$

 # Elasticity of Factor Subsitution (σ):

$Elasticity\_of\_Factor\_Subsitution := \frac{f1 \cdot f2 \cdot (f2 \cdot x2 + f1 \cdot x1)}{(x1 \cdot x2) \cdot (2 \cdot f1 \cdot f2 \cdot f12 - f1 \cdot f1 \cdot f22 - f2 \cdot f2 \cdot f11)}$;

$$Elasticity\_of\_Factor\_Subsitution := ((-0.4\, x1 + 0.4\, x2 + 6)\,(0.4\, x1 - 0.6\, x2 + 9)\,((-0.4\, x1 + 0.4\, x2 + 6)\, x1 + (0.4\, x1 - 0.6\, x2 + 9)\, x2))\Big/\left(x1\, x2\,\left(0.8\,(-0.4\, x1 + 0.4\, x2 + 6)\,(0.4\, x1 - 0.6\, x2 + 9) + 0.6\,(-0.4\, x1 + 0.4\, x2 + 6)^2 + 0.4\,(0.4\, x1 - 0.6\, x2 + 9)^2\right)\right) \tag{15}$$

 # ElasticityOfFS := eval(E_FS, [x1 = 2, x2 = 3]);
 # Enter value of x1 and x2 as desired to calculate elasticity of factor subsitution.

 # Elasticity of Production or Functional Coefficient (ε): Also gives return to scale (constant (= 1) , incresasing (> 1) or decreasing (< 1)).

$E\_x1 := \frac{f1}{APP1}$; ***# Elasticity of production for factor x1. MPP1 divided by APP1.***

$$E\_x1 := \frac{(-0.4\, x1 + 0.4\, x2 + 6)\, x1}{-0.2\, x1^2 + (0.4\, x2 + 6.)\, x1 - 0.3\, x2^2 + 9.\, x2} \tag{16}$$

$E\_x2 := \frac{f2}{APP2}$; ***# Elasticity of production for factor x2. (MPP2 divided by APP2).***

$$E\_x2 := \frac{(0.4\, x1 - 0.6\, x2 + 9)\, x2}{-0.3\, x2^2 + (0.4\, x1 + 9.)\, x2 - 0.2\, x1^2 + 6.\, x1} \tag{17}$$

$Functional\_Coefficient := simplify(E\_x1 + E\_x2)$;
#Function coefficient or elasticity of production for production function.

$$Functional\_Coefficient := \frac{0.6\, x2^2 + (-0.8\, x1 - 9.)\, x2 + 0.4\, x1^2 - 6.\, x1}{0.3\, x2^2 + (-0.4\, x1 - 9.)\, x2 + 0.2\, x1^2 - 6.\, x1} \tag{18}$$

$Fun\_Coeff\_Alt := simplify\left(\frac{(f1 \cdot x1 + f2 \cdot x2)}{y}\right)$;
#Fun_Coeff_Alt and Fun_Coeff are same just calculated in different ways.

$$Fun\_Coeff\_Alt := \frac{0.6\, x2^2 + (-0.8\, x1 - 9.)\, x2 + 0.4\, x1^2 - 6.\, x1}{0.3\, x2^2 + (-0.4\, x1 - 9.)\, x2 + 0.2\, x1^2 - 6.\, x1} \tag{19}$$

 # Homogenous function:

 # if one common value in term of t^r is obtained for *entire equation, the funciton is homogenous of degree* "r" and *has constant proportion of* return to *scale of r . The r value is also equal* to *functional ceofficient* (**ε**). *The degree of homogenity of MPPs* **and** *APPs of homogenous production functions are* "r-1".

$Homogen := simplify(eval(y, [x1 = t \cdot x1, x2 = t \cdot x2]))$;

$$Homogen := \left(-0.2\, x1^2 + 0.4\, x1\, x2 - 0.3\, x2^2\right) t^2 + (6.\, x1 + 9.\, x2)\, t \tag{20}$$

 # Ridgelines x1 and x2:

Ridgeline_x1 := *solve*(*f1* = 0, *x1*); **# MPP1 = 0 and Stage III begins for input x2.**

$$Ridgeline\_x1 := x2 + 15. \quad (21)$$

Ridgeline_x2 := *solve*(*f2* = 0, *x2*); **# MPP2 = 0 and stage III begins for input x1.**

$$Ridgeline\_x2 := 0.667\, x1 + 15. \quad (22)$$

 # General Cost and Profit Function:

vc := *r1·x1* + *r2·x2*; *fc* := *b*; *Cost_fun* := *vc* + *fc*;

#VC = Variable Cost, FC = Fixed Cost, Cost_fun = Total Cost.

$$vc := 8\, x1 + 10\, x2 \quad (23)$$

$$fc := 0 \quad (23)$$

$$Cost\_fun := 8\, x1 + 10\, x2 \quad (23)$$

TVP := *p·y*; *Profit_fun* := *TVP* - *Cost_fun*; *# TVP = Total Value Product, Profit_fun = Profit.*

$$TVP := -2.0\, x1^2 + 4.0\, x1\, x2 - 3.0\, x2^2 + 60\, x1 + 90\, x2$$

$$Profit\_fun := -2.0\, x1^2 + 4.0\, x1\, x2 - 3.0\, x2^2 + 52\, x1 + 80\, x2 \quad (24)$$

Profit Maximization (Without any Constraints):

$profit := Profit\_fun;$

$$profit := -2.0\,x1^2 + 4.0\,x1\,x2 - 3.0\,x2^2 + 52\,x1 + 80\,x2 \tag{25}$$

$pf1 := diff(profit, x1); pf11 := diff(pf1, x1);$ *# pf11 < 0 for profit max.*

$$pf1 := -4.0\,x1 + 4.0\,x2 + 52$$

$$pf11 := -4.0 \tag{26}$$

$pf2 := diff(profit, x2); pf22 := diff(pf2, x2);$ *# pf22 < 0 for profit max.*

$$pf2 := 4.0\,x1 - 6.0\,x2 + 80$$

$$pf22 := -6.0 \tag{27}$$

$pf12 := diff(pf1, x2); pf21 := diff(pf2, x1);$

$$pf12 := 4.0$$

$$pf21 := 4.0 \tag{28}$$

$EP\_p\_x1 := solve(pf1 = 0, x1);$ ***# Expansion path x1***

$$EP\_p\_x1 := 13. + x2 \tag{29}$$

$EP\_p\_x2 := solve(pf2 = 0, x2);$ ***# Expansion path x2***

$$EP\_p\_x2 := 13.3 + 0.667\,x1 \tag{30}$$

$x2s\_profit := simplify(solve((eval(pf2,\ x1 = EP\_p\_x1)) = 0, x2));$

 # X2Star: Profit maximizing level of input x2.

$$x2s\_profit := 66. \tag{31}$$

$x1s\_profit := simplify(eval(EP\_p\_x1, x2 = x2s\_profit));$ ***# X1Star: Profit maximizing level of input x1.***

$$x1s\_profit := 79. \tag{32}$$

$MaxProfOut := simplify(eval(y, [x1 = x1s\_profit, x2 = x2s\_profit]));$

 # Profit Maximizing Level of Output.

$$MaxProfOut := 588. \tag{33}$$

$ProfitStar := simplify(p \cdot (y = MaxProfOut) - r1 \cdot (x1 = x1s\_profit) - r2 \cdot (x2 = x2s\_profit) - b);$

 # ProfitStar: Maximum profit.

$$ProfitStar := -2.0\,x1^2 + (4.0\,x2 + 52.)\,x1 - 3.0\,x2^2 + 80.\,x2 = 4590. \tag{34}$$

$CostStar\_MaxProfit := simplify(eval(Cost\_fun, [x1 = x1s\_profit, x2 = x2s\_profit]));$

 # Cost at Maximum profit.

$$CostStar\_MaxProfit := 1290. \tag{35}$$

$\frac{r1}{f1} = \frac{r2}{f2};$

$$\frac{8}{-0.4\,x1 + 0.4\,x2 + 6} = \frac{10}{0.4\,x1 - 0.6\,x2 + 9} \tag{36}$$

$\frac{r1}{pf1} = \frac{r2}{pf2};$

$$\frac{8}{-4.0\,x1 + 4.0\,x2 + 52} = \frac{10}{4.0\,x1 - 6.0\,x2 + 80} \tag{37}$$

$BHessian\_Profit\_Max := (pf11 \cdot pf22 - pf12 \cdot pf21);$

#SOC condition: Determinant of Boarder Hessian Matrix > 0 for maximization.

$$BHessian\_Profit\_Max := 8.0 \tag{38}$$

$Curvature\_ProfitMax := -\left(\frac{\mathrm{abs}(pf11 \cdot pf22 - pf12 \cdot pf21)}{pf2^3}\right);$

 # Strictly quasi concave in perfect competition.

$$Curvature\_ProfitMax := -\frac{8.0}{(4.0\,x1 - 6.0\,x2 + 80)^3} \quad \textbf{(39)}$$

$Curvature\_ProfitMax\_Value := eval(Curvature\_ProfitMax, [x1 = x1s\_profit, x2 = x1s\_profit]);$

$$Curvature\_ProfitMax\_Value := 0.0000169 \quad \textbf{(40)}$$

 # *Comparative Static of Profit Maximization: In the Bottom of File.*

#DO COMPARATIVE STATIC MANUALLY

 # Cost Minimization subject to Output Constraints (Min C st (yo - y)): Least Cost Combination of Two Factors of Production.

$Cost := Cost\_fun;$

$$Cost := 8\,x1 + 10\,x2 \tag{41}$$

$LC := Cost + \lambda\cdot(yo - y);$ *# λ is lagrangean multiplier.*

$$LC := 8\,x1 + 10\,x2 + \lambda\left(385 + 0.2\,x1^2 - 0.4\,x1\,x2 + 0.3\,x2^2 - 6\,x1 - 9\,x2\right) \tag{42}$$

$LCf1 := diff(LC, x1); LCf11 := diff(LCf1, x1);$ *#FOC and SOC of lagrangean function wrt x1*

$$LCf1 := 8 + \lambda\,(0.4\,x1 - 0.4\,x2 - 6)$$

$$LCf11 := 0.4\,\lambda \tag{43}$$

$LCf2 := diff(LC, x2); LCf22 := diff(LCf2, x2);$ *#FOC and SOC of lagraangean function wrt x2*

$$LCf2 := 10 + \lambda\,(-0.4\,x1 + 0.6\,x2 - 9)$$

$$LCf22 := 0.6\,\lambda \tag{44}$$

$LCf12 := diff(LCf1, x2); LCf21 := diff(LCf2, x1);$
 # Cross differentiation of LCf1 and LCf2 wrt x2 and x1 respectively. Gives interdependence of factors.

$$LCf12 := -0.4\,\lambda$$

$$LCf21 := -0.4\,\lambda \tag{45}$$

$LCF\lambda := diff(LC, \lambda);$ *#FOC of lagraangean function wrt λ.*

$$LCF\lambda := 385 + 0.2\,x1^2 - 0.4\,x1\,x2 + 0.3\,x2^2 - 6\,x1 - 9\,x2 \tag{46}$$

$LCf1\lambda := solve(LCf1, \lambda);$ ***# λ from LCf1.***

$$LCf1\lambda := -\frac{20.}{x1 - 1.\,x2 - 15.} \tag{47}$$

$LCf2\lambda := solve(LCf2, \lambda);$ ***# λ from LCf2.***

$$LCf2\lambda := \frac{50.}{2.\,x1 - 3.\,x2 + 45.} \tag{48}$$

$\frac{r1}{f1} = \frac{r2}{f2};$

$$\frac{8}{-0.4\,x1 + 0.4\,x2 + 6} = \frac{10}{0.4\,x1 - 0.6\,x2 + 9} \tag{49}$$

$\frac{r1}{LCf1} = \frac{r2}{LCf2};$

$$\frac{8}{8 + \lambda\,(0.4\,x1 - 0.4\,x2 - 6)} = \frac{10}{10 + \lambda\,(-0.4\,x1 + 0.6\,x2 - 9)} \tag{50}$$

$EP\_C\_x1 := solve(LCf1\lambda = LCf2\lambda, x1);$ ***#Expansion path X1***

$$EP\_C\_x1 := 1.22\,x2 - 1.67 \tag{51}$$

$EP\_C\_x2 := solve(LCf1\lambda = LCf2\lambda, x2);$ ***#Expansion Path X2***

$$EP\_C\_x2 := 0.818\,x1 + 1.36 \tag{52}$$

$x2s\_cost := solve((eval(LCF\lambda, x1 = EP\_C\_x1)),\ x2);$
#X2Star: Cost Minimizing input x2 demand function. #Use small value

$$x2s\_cost := 120., 30.1 \tag{53}$$

$x1s\_cost := eval(EP\_C\_x1, x2 = x2s\_cost);$

#X1Star: Cost Minimizing input x1 demand function. #use small value.

$$x1s\_cost := (146., 36.7) - 1.67 \tag{54}$$

$CostStar\_Cost := r1 \cdot x1s\_cost + r2 \cdot x2s\_cost + b;$

#CostStar: Minimum Cost for the production of given level of output.

$$CostStar\_Cost := (2370., 595.) - 13.4 \tag{55}$$

$ystar\_cost := eval(y, [x1 = x1s\_cost, x2 = x2s\_cost]);$

#Ystar: Output level produced. This should be equal to given level of output.

$$ystar\_cost := -0.2\, ((146., 36.7) - 1.67)^2 + 0.4\, ((146., 36.7) - 1.67)\, (120., 30.1) - 0.3\, (120., 30.1)^2 + (1960., 491.) - 10.0 \tag{56}$$

$LCf1\lambda Star := eval(LCf1\lambda, [x1 = x1s\_cost, x2 = x2s\_cost]);$

#Lagrangean multiplier λ1 Star #Use positive value

$$LCf1\lambda Star := -\frac{20.}{(26., 6.6) - 16.7} \tag{57}$$

$LCf2\lambda Star := eval(LCf2\lambda, [x1 = x1s\_cost, x2 = x2s\_cost]);$

#Lagrangean multiplier λ2 Star #Use positive value.

$$LCf2\lambda Star := \frac{50.}{(-68., -16.9) + 41.7} \tag{58}$$

$BHessian\_Cost\_Min := simplify(\lambda \cdot (LCf1 \cdot LCf1 \cdot LCf22 - 2 \cdot LCf1 \cdot LCf2 \cdot LCf12 + LCf2 \cdot LCf2 \cdot LCf11));$

#SOC Condition: Minimum Cost = Determinant of Boarder Hessian Matrix < 0. If > 0, then cost is Maximum.

$$BHessian\_Cost\_Min := 0.0319\, \left(4430. + \left(0.998\, x1^2 + (-2.00\, x2 - 30.0)\, x1 + 1.50\, x2^2 - 44.9\, x2 + 3030.\right) \lambda^2 + (-7360. + 39.9\, x1 + 49.9\, x2)\, \lambda\right) \lambda^2 \tag{59}$$

$BHessian\_Cost\_Min\_Value := eval\left(BHessian\_Cost\_Min, \left[x1 = x1s\_cost, x2 = x2s\_cost, \lambda = LCf1\lambda Star\right]\right);$

$$BHessian\_Cost\_Min\_Value := \frac{1}{((26., 6.6) - 16.7)^2} \left(12.8 \left(4430. + \frac{1}{((26., 6.6) - 16.7)^2} \left(400. \left(0.998\, ((146., 36.7) - 1.67)^2 + ((-240., -60.2) - 30.0)\, ((146., 36.7) - 1.67) + 1.50\, (120., 30.1)^2 + (-5390., -1350.) + 3030.\right)\right) - \frac{20.\, (-7430. + (11800., 2960.))}{(26., 6.6) - 16.7}\right)\right) \tag{60}$$

$Curvature\_CostMin := simplify\left(-\left(\frac{\mathrm{abs}(BHessian\_Cost\_Min)}{\lambda \cdot LCf2^3}\right)\right);$

 # Strictly quasi-concave production function or Convex Isoquant; SOC always holds.

$$Curvature\_CostMin := \frac{1}{\lambda\, (-10. + 0.4\, \lambda\, x1 - 0.6\, \lambda\, x2 + 9.\, \lambda)^3} \left(0.0319\, \left|4430. + \left(0.998\, x1^2 + (-2.00\, x2 - 30.0)\, x1 + 1.50\, x2^2 - 44.9\, x2 + 3030.\right) \lambda^2 + (-7360. + 39.9\, x1 \right. \tag{61}$$

$+ 49.9\, x2)\, \lambda| \, |\lambda|^2)$

$Corvature\_costMin\_Value := eval(Curvature\_CostMin, [x1 = x1s\_cost, x2 = x2s\_cost, \lambda = LCf1\lambda Star]);$

$$Corvature\_costMin\_Value := -\left(0.638 \left|4430. + \frac{1}{((26., 6.6) - 16.7)^2} \left(400. \left(0.998\, ((146., 36.7) - 1.67)^2 + ((-240., -60.2) - 30.0)\, ((146., 36.7) - 1.67) + 1.50\, (120., 30.1)^2 + (-5390., -1350.) + 3030.\right)\right) - \frac{20.\, (-7430. + (11800., 2960.))}{(26., 6.6) - 16.7}\right|\right) \Bigg/ \left(((26., 6.6) - 16.7) \left(-10. - \frac{8.0\, ((146., 36.7) - 1.67)}{(26., 6.6) - 16.7} + \frac{12.0\, (120., 30.1)}{(26., 6.6) - 16.7} - \frac{180.}{(26., 6.6) - 16.7}\right)^3\right) \qquad \textbf{(62)}$$

 # Comparative Statics of Conditional Factor Demands (Cost Minimization): At the bottom of file.

 # DO COMPARATIVE STATIC MANUALLY

 # Maximize output subect to budget constraints (Max y st (Co - C)): Gives same result as before:

$Ly := y + \mu \cdot (Co - Cost)$; *#Lagrangean function.*

$$Ly := -0.2\, x1^2 + 0.4\, x1\, x2 - 0.3\, x2^2 + 6\, x1 + 9\, x2 + \mu\, (936 - 8\, x1 - 10\, x2) \tag{63}$$

$Lyf1 := diff(Ly, x1)$; $Lyf11 := diff(Lyf1, x1)$; *#FOC and SOC of Lagrangean function wrt x1.*

$$Lyf1 := -0.4\, x1 + 0.4\, x2 + 6 - 8\, \mu$$

$$Lyf11 := -0.4 \tag{64}$$

$Lyf2 := diff(Ly, x2)$; $Lyf22 := diff(Lyf2, x2)$; *#FOC and SOC of Lagrangean function wrt x2.*

$$Lyf2 := 0.4\, x1 - 0.6\, x2 + 9 - 10\, \mu$$

$$Lyf22 := -0.6 \tag{65}$$

$Lyf12 := diff(Lyf1, x2)$; $Lyf21 := diff(Lyf2, x1)$;
#Cross SOCs of Lyf1 and Lyf2 wrt x2 and x2 respectively.

$$Lyf12 := 0.4$$

$$Lyf21 := 0.4 \tag{66}$$

$Ly\mu := diff(Ly, \mu)$; *# FOC of Lagrangean function wrt μ.*

$$Ly\mu := 936 - 8\, x1 - 10\, x2 \tag{67}$$

$Lyf1\mu := solve(Lyf1 = 0, \mu)$; *#Solve for μ using Lyf1μ.*

$$Lyf1\mu := 0.750 - 0.0500\, x1 + 0.0500\, x2 \tag{68}$$

$Lyf2\mu := solve(Lyf2 = 0, \mu)$; *#Solve for μ using Lyf1μ.*

$$Lyf2\mu := 0.900 + 0.0400\, x1 - 0.0600\, x2 \tag{69}$$

$EP\_Ly\_x1 := solve(Lyf1\mu = Lyf2\mu, x1)$;
 # Equate μ from above two equations and solve for x1. Expansion path x1.

$$EP\_Ly\_x1 := 1.22\, x2 - 1.67 \tag{70}$$

$EP\_Ly\_x2 := solve(Lyf1\mu = Lyf2\mu, x2)$;
 # Equate μ from above two equations and solve for x2. Expansion path x2.

$$EP\_Ly\_x2 := 0.818\, x1 + 1.36 \tag{71}$$

$x2s\_expd := solve((eval(Ly\mu, x1 = EP\_Ly\_x1)), x2)$;
#X2Star: Demand function of input x2 to maximize the output subject to budget constraints. Ordinary Input Demand Function x2.

$$x2s\_expd := 47.9 \tag{72}$$

$x1s\_expd := eval(EP\_Ly\_x1, x2 = x2s\_expd)$;
#X1Star: Demand function of input x1 to maximize the output subject to budget constraints. Ordinary Input Demand Function x1.

$$x1s\_expd := 56.7 \tag{73}$$

$ystar\_expd := eval(y, [x1 = x1s\_expd, x2 = x2s\_expd])$;
#YStar: Supply function. Total output produced under constrained budget.

$$ystar\_expd := 532. \tag{74}$$

$CostStar\_expd := eval(Cost, [x1 = x1s\_expd, x2 = x2s\_expd])$;
#CostStar: Total cost to produce given amount of output.

$$CostStar\_expd := 933. \tag{75}$$

$Ly\mu Star\_Lyf1\mu := eval(Lyf1\mu, [x1 = x1s\_expd, x2 = x2s\_expd])$;
#Lagrangean Multiplier Star is Marginal product (increase in output per unit cost) under constrained budget condition. LyμStar_Lyf1μ = LyμStar_Lyf2μ.

$$Ly\mu Star\_Lyf1\mu := 0.31 \tag{76}$$

$LyμStar\_Lyf1μ := eval(Lyf2μ, [x1 = x1s\_expd, x2 = x2s\_expd]);$

#Lagrangean Multiplier Star (LyμStar_Lyf1μ = LyμStar_Lyf2μ).

$$LyμStar\_Lyf1μ := 0.30 \tag{77}$$

$BHessian\_Output\_Max := simplify(\, μ \cdot (Lyf1 \cdot Lyf1 \cdot Lyf22 - 2 \cdot Lyf1 \cdot Lyf2 \cdot Lyf12 + Lyf2 \cdot Lyf2 \cdot Lyf11));$

#SOC Condition: Determinant of Boarder Hessian Matrix > 0 for Output Maximization.

$$BHessian\_Output\_Max := (-0.048\, x2^2 - 0.032\, x1^2 + 0.064\, x1\, x2 + 1.44\, x2 + 236.\, μ + 0.96\, x1 - 1.28\, x1\, μ - 1.6\, x2\, μ - 142.\, μ^2 - 97.2)\, μ \tag{78}$$

$BHessian\_Output\_Max\_Value := simplify(eval(BHessian\_Output\_Max, [x1 = x1s\_expd, x2 = x2s\_expd, λ = Lyf1μStar]));$

$$BHessian\_Output\_Max\_Value := -142.\, (μ - 0.255)\, (μ - 0.353)\, μ \tag{79}$$

$Curvature\_Output\_Max := simplify\left(-\left(\frac{\mathrm{abs}(BHessian\_Output\_Max)}{μ \cdot Lyf2^3}\right)\right);$

$$Curvature\_Output\_Max := \frac{1}{μ\,(-9. - 0.4\, x1 + 0.6\, x2 + 10.\, μ)^3}\left(142.\, \left|\left(1.00\, μ^2 + (-1.66 + 0.00901\, x1 + 0.0113\, x2)\, μ + 0.000225\, x1^2 + (-0.000451\, x2 - 0.00676)\, x1 + 0.000338\, x2^2 - 0.0101\, x2 + 0.684\right) μ\right|\right) \tag{80}$$

$Corvature\_Output\_Max\_Value := eval(Curvature\_Output\_Max, [x1 = x1s\_expd, x2 = x2s\_expd, λ = Lyf1μStar]);$

$$Corvature\_Output\_Max\_Value := \frac{142.\, \left|(μ^2 - 0.609\, μ + 0.086)\, μ\right|}{μ\,(-3.0 + 10.\, μ)^3} \tag{81}$$

 # Homogenity of Demand Functions: Factor demand functions are homogenous of degree zero.

#` Comparative Statisc of Output Maximization: `

 # Not Demonstrated in Note. DO COMPARATIVE STATIC MANUALLY

 # Comparative Static and Symmetry* : *# DO COMPARATIVE STATIC MANUALLY IF POSSIBLE:

 # Comparative Static of Profit Maximization:# Revisit this section of comparative static.
 # This is giving error when rin with values. Instead of differentiating function, I may have to
use *FOCs* **and** *SOCs* **to** *solve comparative static probelms.*
Own_Price_Effect_d_x1s_profit := *diff* (*x1s_profit*, *r1*); *# Own Price Effect of x1s* **implies** ***dr2 = dp***
= 0; *Also given by* $\left(\frac{f22}{p\cdot(f11\cdot f22 - f12^2)}\right)$ *For maximum profit, Own Price Effect* < 0.
Error, invalid input: diff received 8, which is not valid for its 2nd argument
Own_Price_Effect_d_x2s_profit := *diff* (*x2s_profit*, *r2*); ***# Own Price Effect of x2s Implies dr1 = dp***
= 0; *Also given by* $\left(\frac{f11}{p\cdot(f11\cdot f22 - f12^2)}\right)$
Error, invalid input: diff received 10, which is not valid for its 2nd argument
Cross_Price_Effect_d_x1s_profit := *diff* (*x1s_profit*, *r2*);
#Cross Price Effect implies "dr2 = dp = 0"   OR
 # This is also given by diff (dx2s_profit, x1) . dr1 = dp = 0. · *(See below)*.
 # Also given by $\left(-\frac{f12}{p\cdot(f11\cdot f22 - f12^2)}\right)$ *OR* $\left(-\frac{f21}{p\cdot(f11\cdot f22 - f12^2)}\right)$. ***This gives values***
in *terms of x1* and *x2 which, I think, is* not *right. Confirm* in *Book*.
Error, invalid input: diff received 10, which is not valid for its 2nd argument
Cross_Price_Effect_d_x2s_profit := *diff* (*x2s_profit*, *r1*);
 # Cross Price Effect of x2s implies dr1 = dp = 0
Error, invalid input: diff received 8, which is not valid for its 2nd argument
Output_Price_Effect_d_x1s_Profit := *simplify*(*diff* (*x1s_profit*, *p*));
 # Output price effect implies dr1 = dr2 = 0.
Error, invalid input: diff received 10, which is not valid for its 2nd argument
Output_Price_Effect_d_x2s_Profit := *diff* (*x2s_profit*, *p*); *# Output price effect implies dr1 = dr2 = 0.*
Error, invalid input: diff received 10, which is not valid for its 2nd argument
 # Economic Interdependence of Factors:
 # If diff(x1s_profit, r2) and diff(x1s_profit, r2) are
 # < 0 means two factors are complementary.
 # = 0 means two factors are independent.
 # > 0 means two factors are competitive.

 # Comparative Statics of Conditional Factor Demands (Cost Minimization): What is the relationship between quantities of factors used and factors prices when output is constant?
#DO COMPARATIVE STATIC MANUALLY
Own_Price_Effect_d_x1s_expd :=

$$\frac{-LCf2\cdot LCf2}{LCf1\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)}$$

 # This is equivalent to diff (*x1s_expd*, *r1*). *This value is negative.*
 # Confirm FOCs & SOCs from Lagraaangean Function or Production Function.

$$Own\_Price\_Effect\_d\_x1s\_expd := \left(0.0500\ (10 + \lambda\ (-0.4\ x1 + 0.6\ x2 - 9))^2\ ((26., 6.6)\right. \qquad \textbf{(82)}$$

$$- 16.7))\Big/\Big(-0.8\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,\lambda - 0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,\lambda - 0.4\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))^2\,\lambda\Big)$$

$$Own\_Price\_Effect\_d\_x2s\_expd := \frac{-LCf1\cdot LCf1}{LCf2\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)}$$

 # This is equivalent to diff(x2s_expd, r1). This value is negative . CHECK THIS FORMULA IN BOOK.

$$Own\_Price\_Effect\_d\_x2s\_expd := -\Big(0.0200\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,((-68., -16.9)+41.7)\Big)\Big/\Big(-0.8\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,\lambda - 0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,\lambda - 0.4\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))^2\,\lambda\Big) \quad \textbf{(83)}$$

$$Cross\_Price\_Effect\_d\_x1s\_expd := \frac{LCf1\cdot LCf2}{LCf1\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)}$$

 # This is equivalent to diff(x1s_expd, r2).

$$Cross\_Price\_Effect\_d\_x1s\_expd := -\Big(0.0500\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,((26., 6.6)-16.7)\Big)\Big/\Big(-0.8\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,\lambda - 0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,\lambda - 0.4\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))^2\,\lambda\Big) \quad \textbf{(84)}$$

$$Cross\_Price\_Effect\_d\_x2s\_expd := \frac{LCf2\cdot LCf1}{LCf2\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)};$$

 # This is equivalent to diff(x2s_expd, r1). CHECK THIS FORMULA IN BOOK.
#These two cross price effects should be equal.

$$Cross\_Price\_Effect\_d\_x2s\_expd := \Big(0.0200\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,((-68., -16.9)+41.7)\Big)\Big/\Big(-0.8\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,\lambda - 0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,\lambda - 0.4\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))^2\,\lambda\Big) \quad \textbf{(85)}$$

$$Output\_Effect\_d\_x1s\_expd := \frac{LCf2\cdot LCf12 - LCf1\cdot LCf22}{LCf1\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)};$$

 # This is equivalent to diff(x1s_expd, y). For some reason, I could **not** *run this differentiation.*
 # If this value > 0, then we have normal factor. And, if this value < 0, we have inferior factor.

$$Output\_Effect\_d\_x1s\_expd := -\Big(0.0500\,(-0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,\lambda - 0.4\,\lambda\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9)))\,((26., 6.6)-16.7)\Big)\Big/\Big(-0.8\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))\,\lambda - 0.6\,(8+\lambda\,(0.4\,x1-0.4\,x2-6))^2\,\lambda - 0.4\,(10+\lambda\,(-0.4\,x1+0.6\,x2-9))^2\,\lambda\Big) \quad \textbf{(86)}$$

$$Output\_Effect\_d\_x2s\_expd := \frac{LCf1\cdot LCf21 - LCf2\cdot LCf11}{LCf1\lambda Star\cdot(2\cdot LCf1\cdot LCf2\cdot LCf12 - LCf1\cdot LCf1\cdot LCf22 - LCf2\cdot LCf2\cdot LCf11)};$$

This is equivalent to diff(x2s_expd, y). For some reason, I could **not** run this differentiation.

If this value > 0, then we have normal factor. And, if this value < 0, we have inferior factor.CHECK THIS FORMULA IN BOOK.

$$Output\_Effect\_d\_x2s\_expd := -\left(0.0500\left(-0.4\left(8+\lambda\left(0.4\,x1-0.4\,x2-6\right)\right)\lambda-0.4\,\lambda\left(10+\lambda\left(-0.4\,x1+0.6\,x2-9\right)\right)\right)\left(\left(26.,6.6\right)-16.7\right)\right)\Big/\left(-0.8\left(8+\lambda\left(0.4\,x1-0.4\,x2-6\right)\right)\left(10+\lambda\left(-0.4\,x1+0.6\,x2-9\right)\right)\lambda-0.6\left(8+\lambda\left(0.4\,x1-0.4\,x2-6\right)\right)^2\lambda-0.4\left(10+\lambda\left(-0.4\,x1+0.6\,x2-9\right)\right)^2\lambda\right) \tag{87}$$

#\` Comparative Statisc of Output Maximization: \`

 # Not Demonstrated in Note. DO COMPARATIVE STATIC MANUALLY

 # HW3_QI_c: Least combination of labor and capital to produce 385 unit of output.

 # Lesat_Combn_Labor ≔ x1s_cost; ***#Chose smallest positive value***

 # Least_combn_capital ≔ x2s_cost; ***#Chose Smallest positive value.***

 # Least_Cost ≔ CostStar_Cost; ***#Chose smallest positive value.***

 # Output_Least_Cost ≔ (ystar_cost);

#Use smallest values inside parenthesis and calculate the cost. This value should be equal or smaller than given unit of output.

 # HW3_QI_d: Appriximate estimated increase in production cost due to unit increase in output:

#LCf1λStar and LCf1λStar are equal.

 # Increase_Cost_Per_Unit_Increae_In_Production ≔ LCf1λStar ;

 # Increase_Cost_Per_Unit_Increae_In_Production ≔ LCf2λStar;

***HW3_Q1_e : Optimum production and input levels :***

 # Labor_Input_for_Optimum_Prod ≔ x1s_expd; # Capital_Input_for_Optimum_Prod ≔ x2s_expd;

 # Optimum_production ≔ ystar_expd;

 # Optimum_Prod_Cost ≔ CostStar_expd;

 # HW3 _Q1_f : Approximate estimated increase in output per unit cost: (LyμStar_Lyf1μ should be equal to LyμStar_Lyf2μ).

 # Estimated_Output_increase_per_unit_cost ≔ LyμStar_Lyf1μ;

 # Estimated_Output_increase_per_unit_cost ≔LyμStar_Lyf1μ;