Workshop for Economic Modeling in Julia

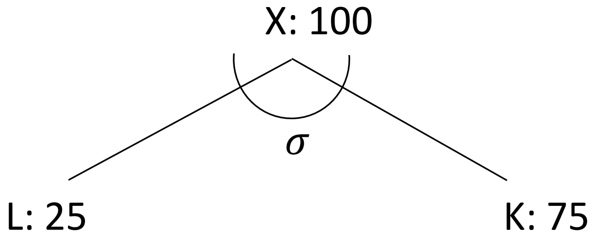
Y.-H. Henry Chen
Research Scientist, MIT CS3

Feb 3, 2026

Center for
Sustainability Science
and Strategy

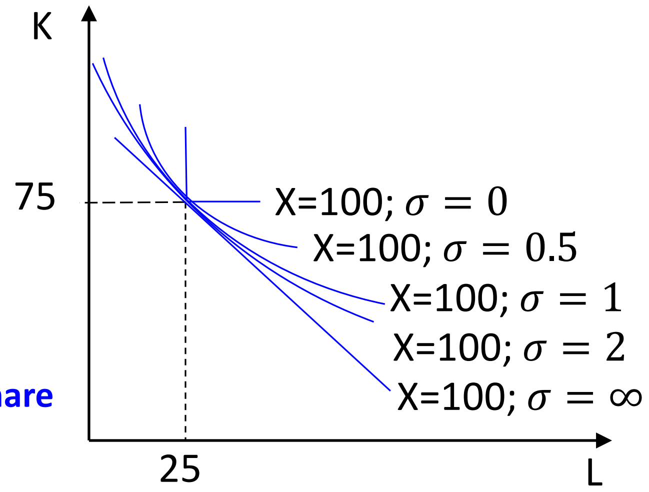
Production technology

- Constant elasticity of substitution (CES) function



$$\sigma \equiv \left[\frac{d\left(\frac{F_l}{F_k}\right)}{\left(\frac{F_l}{F_k}\right)}\right] / \left[\frac{d\left(\frac{P_k}{P_l}\right)}{\left(\frac{P_k}{P_l}\right)}\right] = \frac{1}{1-\rho}$$

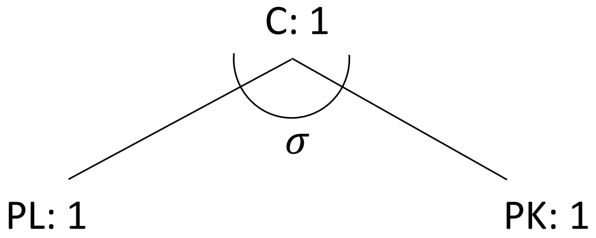
γ_m: cost share



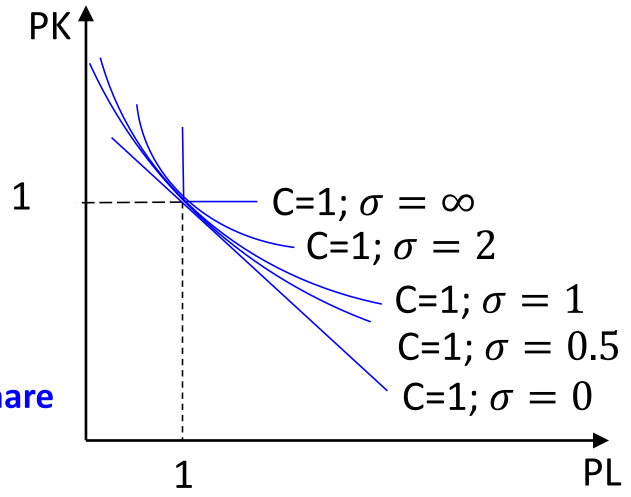
$$\begin{cases} Q = \bar{Q}\sum_m \gamma_m \cdot \frac{F_m}{\bar{F}_m} \quad ; \quad m = L, K & \sigma = \infty \\ Q = \bar{Q}\left[\sum_m \gamma_m \cdot \left(\frac{F_m}{\bar{F}_m}\right)^{\rho}\right]^{\frac{1}{\rho}} & \sigma > 0 \quad ; \quad \sigma \neq 1 \\ Q = \bar{Q}\prod_m \left(\frac{F_m}{\bar{F}_m}\right)^{\gamma_m} & \sigma = 1 \\ Q = \bar{Q}\min\left[\frac{F_L}{\bar{F}_L}, \frac{F_K}{\bar{F}_K}\right] & \sigma = 0 \end{cases}$$

Production technology

- Duality theorem: production function => cost function



$$\sigma \equiv [\frac{d\left(\frac{F_l}{F_k}\right)}{\left(\frac{F_l}{F_k}\right)}]/[\frac{d\left(\frac{P_k}{P_l}\right)}{\left(\frac{P_k}{P_l}\right)}] = \frac{1}{1-\rho}$$



γ_m: cost share

$$\begin{cases} C = \bar{C}\min\left[\frac{P_L}{\bar{P}_L}, \frac{P}{\bar{P}_K}\right]; \quad m = L, K & \sigma = \infty \\ C = \bar{C}\left[\sum_m \gamma_m \cdot \left(\frac{P_m}{\bar{P}_m}\right)^{1-\sigma}\right]^{\frac{1}{1-\sigma}} & \sigma > 0 \quad ; \quad \sigma \neq 1 \\ C = \bar{C}\prod_m \left(\frac{P_m}{\bar{P}_m}\right)^{\gamma_m} & \sigma = 1 \\ C = \bar{C}\sum_m \gamma_m \cdot \left(\frac{P_m}{\bar{P}_m}\right) & \sigma = 0 \end{cases}$$

Social Accounting Matrix

- M21_mcp.gms; M21.gms (Markusen, 2002)
- Summarizing the input-output structure of an economy

	Production Sectors			Consumers
Markets	X	Y	W	CONS
PX	100		-100	
PY		100	-100	
PW			200	-200
PL	-25	-75		100
PK	-75	-25		100

Social Accounting Matrix

		Production Sectors			Consumers
Markets	X	Y	W		CONS
PX	100		-100		
PY		100	-100		
PW			200		-200
PL	-25	-75			100
PK	-75	-25			100

Zero-profit condition for X : MC-MR >= 0 ; Q >= 0 ; (MC-MR)*Q = 0 ; No need to worry about MC-MR < 0 (Why?)

Social Accounting Matrix

Market clearing condition for X : S-D >= 0 ; P >= 0 ; (S-D)*P = 0 ; No need to worry about S-D < 0 (why?)

	Production Sectors			Consumers
Markets	X	Y	W	CONS
PX	100		-100	
PY		100	-100	
PW			200	-200
PL	-25	-75		100
PK	-75	-25		100

Zero-profit condition for X : MC-MR >= 0 ; Q >= 0 ; (MC-MR)*Q = 0 ; No need to worry about MC-MR < 0 (Why?)

Social Accounting Matrix

Market clearing condition for X : S-D >= 0 ; P >= 0 ; (S-D)*P = 0 ; No need to worry about S-D < 0 (why?)

	Production Sectors			Consumers
Markets	X	Y	W	CONS
PX	100		-100	
PY		100	-100	
PW			200	-200
PL	-25	-75		100
PK	-75	-25		100

Income balance condition : Income = Expenditure

Zero-profit condition for X : MC-MR >= 0 ; Q >= 0 ; (MC-MR)*Q = 0 ; No need to worry about MC-MR < 0 (Why?)

Model

An excerpt for M21_mcp.gms (Markusen, 2002)

```
*        Zero profit inequalities

PRF_X..          100 * PL**0.25 * PK**0.75 * (1+TX) =G= 100*PX;

PRF_Y..          100 * PL**0.75* PK**0.25 =G= 100*PY;

PRF_W..          200 * PX**0.5 * PY**0.5 =G= 200*PW;

*        Market clearance inequalities

MKT_X..          100 * X =G= 100 * W * PX**0.5 * PY**0.5 / PX;

MKT_Y..          100 * Y =G= 100 * W * PX**0.5 * PY**0.5 / PY;

MKT_W..          200 * W =E= CONS / PW;

MKT_L..          100 * LENDOW =G= 25 * X * PL**0.25 * PK**0.75 / PL +
                                  75 * Y * PL**0.75 * PK**0.25 / PL;

MKT_K..          100 =G= 75 * X * PL**0.25 * PK**0.75 / PK +
                         25 * Y * PL**0.75 * PK**0.25 / PK;

*        Income balance equations

I_CONS..         CONS =E= 100*LENDOW*PL + 100*PK + TX*100*X*PL**0.25*PK**0.75;

MODEL ALGEBRAIC /PRF_X.X, PRF_Y.Y, PRF_W.W, MKT_X.PX, MKT_Y.PY, MKT_L.PL, MKT_K.PK, MKT_W.PW, I_CONS.CONS /;

*        Numeraire

PW.FX = 1;
```

Model

- Let us check the Julia version: M21_mcp.jl
- https://chenyhmitedu.github.io/docs/M21_mcp.jl

Model

An excerpt for M21.gms (Markusen, 2002)

```
$ONTEXT
$MODEL:M21

$SECTORS:
        X       ! Activity level for sector X
        Y       ! Activity level for sector Y
        W       ! Activity level for sector W (Hicksian welfare index)

$COMMODITIES:
        PX      ! Price index for commodity X
        PY      ! Price index for commodity Y
        PL      ! Price index for primary factor L
        PK      ! Price index for primary factor K
        PW      ! Price index for welfare (expenditure function)

$CONSUMERS:
        CONS    ! Income level for consumer CONS

$PROD:X s:1
        O:PX  Q:100
        I:PL  Q:25 A:CONS T:TX
        I:PK  Q:75 A:CONS T:TX
```

```
$PROD:Y s:1
        O:PY   Q:100
        I:PL   Q:75
        I:PK   Q:25

$PROD:W s:1
        O:PW    Q:200
        I:PX    Q:100
        I:PY    Q:100

$DEMAND:CONS
        D:PW    Q:200
        E:PL    Q:(100*LENDOW)
        E:PK    Q:100

$OFFTEXT
$SYSINCLUDE mpsgeset M21
PW.FX = 1;
```

Model

- Let us check the Julia version: M21.jl
- https://chenyhmitedu.github.io/docs/M21.jl

Project

- For the two models, enhance code reusability and avoid repetition

  - Sectors are stored in vectors
  - SAM values are stored in dictionaries
  - Replace numbers in code blocks by key-specified dictionaries

Project

Hint:

```
 |  |  |  |Production Sectors          Consumers
   Markets     | D("A")  D("B )  D("W")  |         CONS
   -------------------------------------------------------
    P("X")     |  80      20     -100    |
    P("Y")     |  20      80     -100    |
    P("W")     |                  200    |         -200
    P("L")     | -40     -60             |          100
    P("K")     | -60     -40             |          100
   -------------------------------------------------------

=#

# Define sectors and factors

C = [:X, :Y]
W = [:W]
F = [:L, :K]
S = [:A, :B]

I = S ∪ W
G = C ∪ W ∪ F

# I/O data & elasticities

out0                = Dict((row, col) => 0 for row ∈ C ∪ W, col ∈ I)
in0                 = Dict((row, col) => 0 for row ∈ C ∪ F, col ∈ I)
end0                = Dict(row => 0 for row ∈ F)
te0                 = Dict(row => 0.0 for row ∈ S)

```

```
out0[(:X, :A)]   = 80
out0[(:Y, :A)]   = 20
out0[(:X, :B)]   = 20
out0[(:Y, :B)]   = 80
out0[(:W, :W)]   = 200
in0[(:L, :A)]    = 40
in0[(:K, :A)]    = 60
in0[(:L, :B)]    = 60
in0[(:K, :B)]    = 40
in0[(:X, :W)]    = 100
in0[(:Y, :W)]    = 100
end0[:L]         = 100
end0[:K]         = 100
te0[:A]          = 2.0
te0[:B]          = 1.5

M23 = MPSGEModel()

@parameters(M23, begin
    TA[S], 0
end)

@sectors(M23, begin
    D[I]
end)

@commodities(M23, begin
    P[G]
end)
```

Bibliography

- Markusen, J. (2002). General-Equilibrium Modeling using GAMS and MPS/GE: Some Basics. University of Colorado, Boulder. https://www.mpsge.org/tutorial.pdf
- Rutherford, T. (2002). Lecture Notes on Constant Elasticity Functions. University of Colorado, Boulder. https://downloads.regulations.gov/EPA-HQ-OAR-2022-0730-0088/attachment_58.pdf