

# Capítulo 1: Naturaleza de la Economía Matemática

Fundamentos Metodológicos y Epistemológicos

Emanuel Quintana Silva

4 de diciembre de 2025

## Tabla de contenidos

### 1 Naturaleza de la Economía Matemática

#### 1.1 Introducción

La economía matemática representa uno de los desarrollos metodológicos más significativos en la ciencia económica del siglo XX (?). Su evolución desde los trabajos pioneros de Cournot (1838) hasta la formalización axiomática de Debreu (?) ha transformado radicalmente la manera en que comprendemos y analizamos los fenómenos económicos.

Este capítulo establece los fundamentos conceptuales que distinguen la economía matemática como **método analítico**, no como rama sustantiva de la economía. Exploraremos su alcance, metodología, y la relación dialéctica con enfoques complementarios como la econometría y el análisis cualitativo (?).

##### 1.1.1 Objetivos de Aprendizaje

Al finalizar este capítulo, el estudiante será capaz de:

1. **Comprender** la definición formal y el alcance metodológico de la economía matemática
2. **Distinguir** entre economía matemática y economía literaria, identificando ventajas comparativas
3. **Diferenciar** economía matemática de econometría, reconociendo su complementariedad
4. **Analizar críticamente** las ventajas y limitaciones del formalismo matemático
5. **Evaluar** la relación entre teoría matemática, análisis empírico e intuición económica
6. **Aplicar** herramientas computacionales para implementar modelos matemáticos

##### 1.1.2 Contexto Histórico

La matematización de la economía tiene raíces profundas:

###### Hitos Históricos en Economía Matemática

- **1838:** Antoine Augustin Cournot - *Recherches sur les principes mathématiques de la théorie des richesses*

- **1874:** Léon Walras - Teoría del equilibrio general
- **1881:** Francis Ysidro Edgeworth - *Mathematical Psychics*
- **1947:** Paul Samuelson - *Foundations of Economic Analysis*
- **1959:** Gerard Debreu - Formalización axiomática del equilibrio general
- **1972:** Kenneth Arrow y Gerard Debreu - Premio Nobel por teoría del equilibrio general

## 1.2 1.1 Definición y Alcance

### 1.2.1 1.1.1 Naturaleza Metodológica

La economía matemática no constituye una subdisciplina económica (como la economía laboral o la economía del desarrollo), sino un **enfoque metodológico transversal** aplicable a cualquier área del análisis económico (?).

#### ! Definición 1.1 (Economía Matemática)

La **economía matemática** es un método de análisis económico caracterizado por:

1. **Formulación simbólica:** Uso de notación matemática para expresar relaciones económicas
2. **Razonamiento formal:** Aplicación de teoremas y técnicas matemáticas establecidas
3. **Estructuras axiomáticas:** Desarrollo deductivo desde premisas explícitas hacia conclusiones lógicas
4. **Generalidad abstracta:** Capacidad de analizar sistemas de  $n$  dimensiones

Formalmente, si  $\mathcal{E}$  denota el conjunto de proposiciones económicas y  $\mathcal{M}$  el conjunto de estructuras matemáticas, la economía matemática establece un **homomorfismo**:

$$\phi : \mathcal{E} \rightarrow \mathcal{M}$$

que preserva las relaciones lógicas entre proposiciones económicas.

### 1.2.2 1.1.2 Características Fundamentales

Tabla 1: Características distintivas de la economía matemática

Característica	Descripción	Ejemplo
Lenguaje	Notación simbólica formal (variables, funciones, operadores)	$U : \mathbb{R}_+^n \rightarrow \mathbb{R}$
Precisión	Definiciones inequívocas y relaciones exactas	$\frac{\partial U}{\partial x_i} > 0 \forall i$
Dimensionalidad	Capacidad de manejar espacios de alta dimensión	$\max_{x \in \mathbb{R}_+^n} U(x) \text{ s.a. } p \cdot x \leq m$
Supuestos	Explicitación completa de premisas y condiciones	Continuidad, diferenciabilidad, convexidad
Derivación	Proceso deductivo riguroso mediante teoremas	Teorema de Kuhn-Tucker
Generalización	Resultados aplicables a clases amplias de problemas	Teorema del sobre

### 1.2.3 1.1.3 Ejemplo Introductorio: Función de Demanda

Consideremos la representación matemática de una relación de demanda:

```
import numpy as np
import matplotlib.pyplot as plt
import sympy as sp

# Configuración de estilo
plt.style.use('seaborn-v0_8-darkgrid')
plt.rcParams['figure.dpi'] = 100
plt.rcParams['font.size'] = 10

# 1. Representación simbólica
P, Q, a, b = sp.symbols('P Q a b', positive=True, real=True)
demanda_simbolica = sp.Eq(Q, a - b*P)

print("*"*60)
print("REPRESENTACIÓN MATEMÁTICA DE LA DEMANDA")
print("*"*60)
print("\n1. FORMA EXPLÍCITA (Cantidad como función del precio):")
print(f" {demanda_simbolica}")
print("\n2. FORMA INVERSA (Precio como función de la cantidad):")
P_inverso = sp.solve(demanda_simbolica, P)[0]
print(f" P = {P_inverso}")
print("\n3. FORMA IMPLÍCITA (Relación general):")
print(f" F(P,Q) = {demanda_simbolica.lhs - demanda_simbolica.rhs} = 0")

# 2. Visualización numérica
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Parámetros numéricos
a_val, b_val = 100, 2
P_vals = np.linspace(0, 50, 100)
Q_vals = np.maximum(0, a_val - b_val * P_vals)

# Panel A: Forma estándar (P en eje vertical)
ax1 = axes[0]
ax1.plot(Q_vals, P_vals, 'b-', linewidth=2.5, label='$P = \frac{100-Q}{2}$')
ax1.fill_between(Q_vals, 0, P_vals, alpha=0.1, color='blue')
ax1.set_xlabel('Cantidad (Q)', fontsize=11, fontweight='bold')
ax1.set_ylabel('Precio (P)', fontsize=11, fontweight='bold')
ax1.set_title('Representación Estándar\n(Curva de Demanda)',
              fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend(loc='upper right', fontsize=10)
ax1.set_xlim(0, 105)
ax1.set_ylim(0, 55)
ax1.axhline(y=0, color='k', linewidth=0.5)
```

```

ax1.axvline(x=0, color='k', linewidth=0.5)

# Panel B: Forma explícita (Q como función de P)
ax2 = axes[1]
ax2.plot(P_vals, Q_vals, 'r-', linewidth=2.5, label='$Q = 100 - 2P$')
ax2.fill_between(P_vals, 0, Q_vals, alpha=0.1, color='red')
ax2.set_xlabel('Precio (P)', fontsize=11, fontweight='bold')
ax2.set_ylabel('Cantidad (Q)', fontsize=11, fontweight='bold')
ax2.set_title('Forma Explícita\n(Función de Demanda)', fontsize=12, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend(loc='upper right', fontsize=10)
ax2.set_xlim(0, 55)
ax2.set_ylim(0, 105)
ax2.axhline(y=0, color='k', linewidth=0.5)
ax2.axvline(x=0, color='k', linewidth=0.5)

# Panel C: Elasticidad
P_elast = P_vals[P_vals > 0]
Q_elast = a_val - b_val * P_elast
elasticidad = -(b_val * P_elast / Q_elast)

ax3 = axes[2]
ax3.plot(Q_elast, elasticidad, 'g-', linewidth=2.5,
          label='$\backslash varepsilon = -\frac{dQ}{dP} \frac{P}{Q}$')
ax3.axhline(y=-1, color='red', linestyle='--', alpha=0.7,
          label='Elasticidad unitaria')
ax3.fill_between(Q_elast, elasticidad, -1,
                  where=(elasticidad < -1), alpha=0.1, color='blue',
                  label='Elástica $|varepsilon| > 1$')
ax3.fill_between(Q_elast, elasticidad, -1,
                  where=(elasticidad >= -1), alpha=0.1, color='red',
                  label='Inelástica $|varepsilon| < 1$')
ax3.set_xlabel('Cantidad (Q)', fontsize=11, fontweight='bold')
ax3.set_ylabel('Elasticidad-Precio', fontsize=11, fontweight='bold')
ax3.set_title('Elasticidad de la Demanda\n(Concepto Derivado)', fontsize=12, fontweight='bold')
ax3.grid(True, alpha=0.3)
ax3.legend(loc='lower right', fontsize=9)
ax3.set_xlim(0, 105)
ax3.set_ylim(-5, 0)

plt.tight_layout()
plt.show()

print("\n4. PROPIEDADES MATEMÁTICAS DERIVADAS:")
print(f"    • Pendiente: dQ/dP = -{b_val}")

```

```

print(f" • Intercepto vertical: P-máx = {a_val/b_val}")
print(f" • Intercepto horizontal: Q-máx = {a_val}")
print(f" • Elasticidad en punto medio: $\varepsilon = -b * (a/b)/2$ / (a/b/2)
=====
```

## REPRESENTACIÓN MATEMÁTICA DE LA DEMANDA

### 1. FORMA EXPLÍCITA (Cantidad como función del precio):

$$Q(P) = -\frac{P}{b} + a$$

### 2. FORMA INVERSA (Precio como función de la cantidad):

$$P(Q) = -\frac{Q}{b} + a$$

### 3. FORMA IMPLÍCITA (Relación general):

$$F(P, Q) = P \cdot b + Q - a = 0$$

### 4. PROPIEDADES MATEMÁTICAS DERIVADAS:

- Pendiente:  $dQ/dP = -b$
- Intercepto vertical:  $P_{máx} = a/b$
- Intercepto horizontal:  $Q_{máx} = a$
- Elasticidad en punto medio:  $\varepsilon = -b * (a/b)/2 / (a/b/2)$

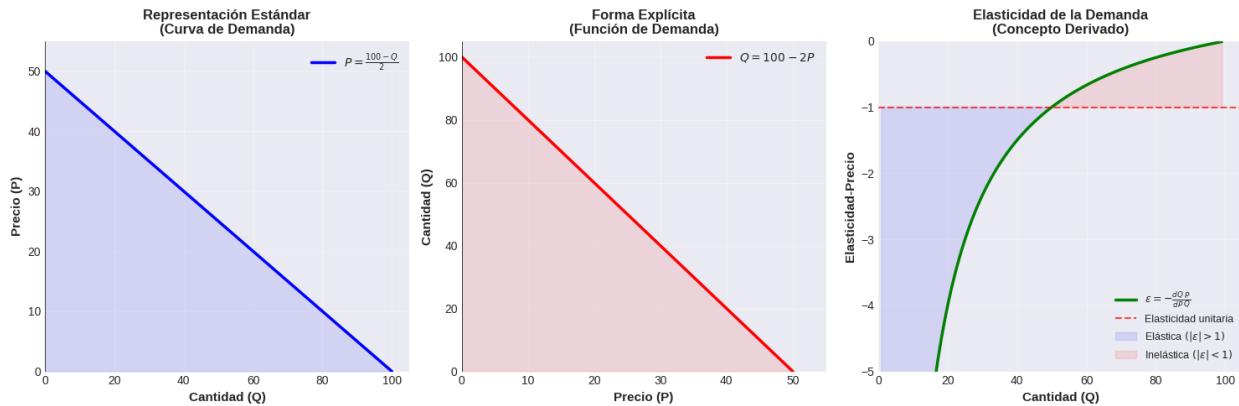


Figura 1: Representaciones alternativas de la función de demanda

### 💡 Interpretación Económica

La formulación matemática  $Q = a - bP$  con  $a, b > 0$  codifica dos principios económicos:

1. **Ley de la demanda:**  $\frac{\partial Q}{\partial P} = -b < 0$  (relación inversa)
2. **Demanda finita:**  $\lim_{P \rightarrow \infty} Q(P) = -\infty$  implica  $Q = 0$  para  $P \geq a/b$

La ventaja del formalismo es que estas propiedades se derivan **lógicamente** de la especificación funcional, no por intuición.

#### 1.2.4 1.1.4 Alcance Aplicativo

La economía matemática se aplica transversalmente:

Tabla 2: Áreas de aplicación de la economía matemática

Área	Problemas Típicos	Herramientas Matemáticas
Microeconomía	Optimización del consumidor/productor, equilibrio competitivo	Cálculo multivariado, optimización convexa
Macroeconomía	Modelos de crecimiento, ciclos económicos, política monetaria	Ecuaciones diferenciales, control óptimo
Economía Pública	Incidencia tributaria, provisión de bienes públicos	Teoría de juegos, análisis de bienestar
Economía Internacional	Ventaja comparativa, flujos de capital, tipos de cambio	Modelos de equilibrio general, comercio
Economía Financiera	Valoración de activos, gestión de riesgos, equilibrio de portafolio	Procesos estocásticos, ecuaciones diferenciales estocásticas
Economía del Desarrollo	Trampa de pobreza, acumulación de capital, demografía	Sistemas dinámicos, modelos de sobrelapamiento generacional
Economía Ambiental	Externalidades, recursos comunes, sostenibilidad	Teoría de juegos, programación dinámica
Economía Laboral	Búsqueda de empleo, capital humano, discriminación	Búsqueda y matching, modelos principal-agente

### 1.3 1.2 Economía Matemática vs. Economía Literaria

#### 1.3.1 1.2.1 Naturaleza Común del Análisis Teórico

Tanto el enfoque matemático como el literario comparten el objetivo fundamental de la **teoría económica deductiva (?)**:

**i** Objetivo Compartido

“Derivar un conjunto de conclusiones o teoremas a partir de un conjunto dado de axiomas o hipótesis mediante un proceso de razonamiento lógicamente válido”

La diferencia radica en el **medio de expresión** y las **herramientas de inferencia**, no en la naturaleza del análisis.

#### 1.3.2 1.2.2 Comparación Sistemática

Tabla 3: Comparación metodológica: Economía matemática vs. literaria

Aspecto	Economía Matemática	Economía Literaria
Lenguaje de formulación	Símbolos, ecuaciones, espacios abstractos	Palabras, enunciados verbales
Proceso de razonamiento	Teoremas matemáticos, demostraciones formales	Lógica discursiva, argumentación

Tabla 3: Comparación metodológica: Economía matemática vs. literaria

Aspecto	Economía Matemática	Economía Literaria
Precisión conceptual	Alta: definiciones inequívocas	Variable: admite ambigüedades
Capacidad de generalización	Caso general de $n$ variables	Limitada por dimensionalidad
Explicitación de supuestos	Completa y necesaria	Puede ser implícita
Verificabilidad de derivaciones	Alta: pasos verificables	Baja: interpretaciones múltiples
Accesibilidad inicial	Requiere formación matemática	Más intuitiva inicialmente
Potencia analítica	Muy alta para problemas complejos	Limitada en alta complejidad

### 1.3.3 1.2.3 Ventajas Comparativas del Formalismo Matemático

#### 1.3.3.1 Ventaja 1: Concisión y Precisión

El lenguaje matemático elimina ambigüedades semánticas inherentes al lenguaje natural.

**Ejemplo 1.1** (Utilidad marginal decreciente)

- **Formulación literaria:** “La satisfacción adicional obtenida del consumo de una unidad adicional de un bien disminuye a medida que se consume más del bien, manteniendo constante el consumo de otros bienes”
- **Formulación matemática:**

$$\frac{\partial^2 U}{\partial x_i^2} < 0 \quad \forall i \in \{1, \dots, n\}$$

La formulación matemática es: - **Más breve:** Una línea vs. un párrafo - **Más precisa:** No admite interpretaciones alternativas - **Más general:** Aplica a  $n$  bienes simultáneamente

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure(figsize=(16, 5))

# Panel A: Función de utilidad total
ax1 = fig.add_subplot(131)
x = np.linspace(0.1, 10, 100)
U = np.log(x) # Utilidad logarítmica (cóncava)
ax1.plot(x, U, 'b-', linewidth=2.5, label='$U(x) = \ln(x)$')
ax1.set_xlabel('Cantidad consumida (x)', fontsize=11, fontweight='bold')
ax1.set_ylabel('Utilidad total U(x)', fontsize=11, fontweight='bold')
ax1.set_title('Función de Utilidad\n(Cóncava)', fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend(fontsize=10)
```

```

# Panel B: Utilidad marginal
ax2 = fig.add_subplot(132)
U_marginal = 1/x # Primera derivada
ax2.plot(x, U_marginal, 'r-', linewidth=2.5,
          label='$U'(x) = \frac{1}{x}')
ax2.set_xlabel('Cantidad consumida (x)', fontsize=11, fontweight='bold')
ax2.set_ylabel('Utilidad marginal $U'(x)$', fontsize=11, fontweight='bold')
ax2.set_title('Utilidad Marginal\n(Decreciente)', fontsize=12, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend(fontsize=10)
ax2.set_ylim(0, 2)

# Panel C: Curvatura (segunda derivada)
ax3 = fig.add_subplot(133)
U_segunda = -1/(x**2) # Segunda derivada
ax3.plot(x, U_segunda, 'g-', linewidth=2.5,
          label='$U''(x) = -\frac{1}{x^2} < 0$')
ax3.axhline(y=0, color='black', linestyle='--', linewidth=1, alpha=0.5)
ax3.fill_between(x, U_segunda, 0, alpha=0.2, color='green')
ax3.set_xlabel('Cantidad consumida (x)', fontsize=11, fontweight='bold')
ax3.set_ylabel('Curvatura $U''(x)$', fontsize=11, fontweight='bold')
ax3.set_title('Segunda Derivada\n(Cóncava: $U''<0$)',
              fontsize=12, fontweight='bold')
ax3.grid(True, alpha=0.3)
ax3.legend(fontsize=10)
ax3.set_ylim(-1, 0.1)

plt.tight_layout()
plt.show()

print("VERIFICACIÓN ANALÍTICA:")
print(f"  U(x) = ln(x)")
print(f"  U'(x) = 1/x > 0  (utilidad marginal positiva)")
print(f"  U''(x) = -1/x² < 0  (utilidad marginal decreciente)")

```

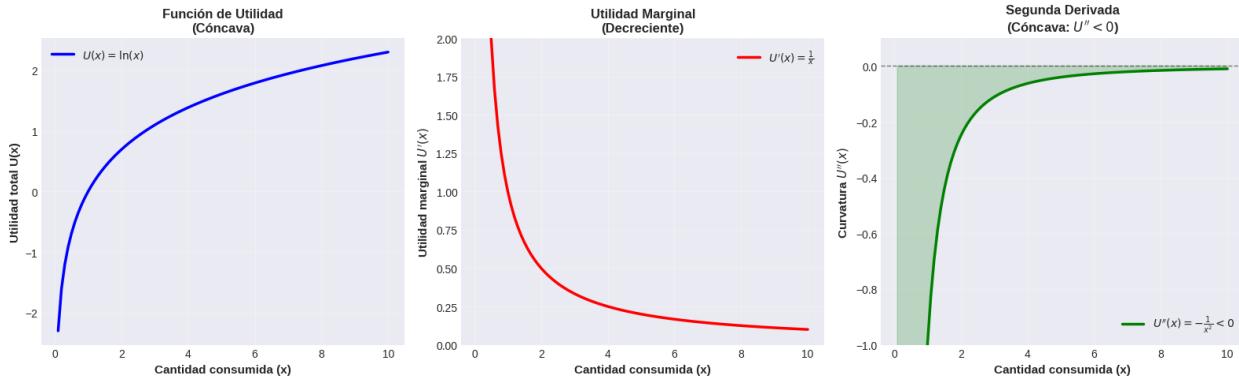


Figura 2: Utilidad marginal decreciente: representación matemática

#### VERIFICACIÓN ANALÍTICA:

$$U(x) = \ln(x)$$

$U'(x) = 1/x > 0$  (utilidad marginal positiva)

$U'''(x) = -1/x^2 < 0$  (utilidad marginal decreciente)

#### 1.3.3.2 Ventaja 2: Arsenal de Teoremas Disponibles

La matemática proporciona herramientas listas para aplicar (?):

! Teorema 1.1 (Condiciones de Primer Orden para Optimización)

Sea  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  una función diferenciable. Si  $x^* \in \text{int}(\mathbb{R}_+^n)$  es un máximo local de  $f$ , entonces necesariamente:

$$\nabla f(x^*) = \mathbf{0} \quad (\text{condiciones de primer orden})$$

Si además  $f$  es cóncava, esta condición es también **suficiente** para un máximo global.

**Aplicación económica inmediata:** Maximización de utilidad del consumidor

```
from scipy.optimize import minimize
import sympy as sp

# Parámetros
alpha1, alpha2 = 0.6, 0.4
p1, p2, m = 2, 3, 100 # precios e ingreso

# Función de utilidad Cobb-Douglas
def utilidad(x):
    return -(x[0]**alpha1 * x[1]**alpha2) # Negativo para minimizar

# Restricción presupuestaria
def restriccion(x):
    return p1*x[0] + p2*x[1] - m

# Optimización numérica
```

```

from scipy.optimize import minimize
cons = {'type': 'eq', 'fun': restriccion}
x0 = [10, 10]
bounds = [(0.1, None), (0.1, None)]
resultado = minimize(utilidad, x0, method='SLSQP', bounds=bounds, constraints=cons)
x1_opt, x2_opt = resultado.x
U_opt = -resultado.fun

# Visualización
fig, axes = plt.subplots(1, 2, figsize=(14, 6))

# Panel izquierdo: Problema de optimización
ax1 = axes[0]
x1_vals = np.linspace(0, 60, 100)
x2_restriccion = (m - p1*x1_vals) / p2

# Curvas de indiferencia
U_levels = [U_opt * 0.6, U_opt * 0.8, U_opt, U_opt * 1.1]
for U_level in U_levels:
    x2_curva = (U_level / (x1_vals**alpha1))**(1/alpha2)
    if U_level == U_opt:
        ax1.plot(x1_vals, x2_curva, 'r-', linewidth=3,
                  label=f'Curva óptima $U^* = {U_opt:.2f}$', zorder=3)
    else:
        ax1.plot(x1_vals, x2_curva, 'b--', linewidth=1, alpha=0.5)

ax1.plot(x1_vals, x2_restriccion, 'g-', linewidth=2.5,
          label=f'Restricción: ${p1}x_1 + {p2}x_2 = {m}$')
ax1.scatter(x1_opt, x2_opt, color='red', s=200, zorder=5, marker='*',
            label=f'Óptimo: ${x^* = ({x1_opt:.2f}, {x2_opt:.2f})}$')
ax1.set_xlabel('Bien 1 $(x_1)$', fontsize=11, fontweight='bold')
ax1.set_ylabel('Bien 2 $(x_2)$', fontsize=11, fontweight='bold')
ax1.set_title('Maximización del Consumidor\n(Representación Gráfica)',
              fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend(fontsize=9, loc='upper right')
ax1.set_xlim(0, 55)
ax1.set_ylim(0, 35)

# Panel derecho: Verificación de CPO
ax2 = axes[1]
ax2.axis('off')

# Texto explicativo
texto_cpo = f"""
VERIFICACIÓN ANALÍTICA DE CPO

```

Problema:

```
max U(x1,x2) = x1^{alpha1} * x2^{alpha2}
s.a. {p1}x1 + {p2}x2 = {m}
```

Lagrangiano:

```
L = x1^{alpha1} * x2^{alpha2} + lambda*({m} - {p1}x1 - {p2}x2)
```

Condiciones de primer orden:

```
dL/dx1 = {alpha1}*x1^{alpha1-1}*x2^{alpha2} - lambda*p1 = 0
dL/dx2 = {alpha2}*x1^{alpha1}*x2^{alpha2-1} - lambda*p2 = 0
dL/dlambda = {m} - {p1}x1 - {p2}x2 = 0
```

Solución óptima:

```
x1* = {x1_opt:.4f}
x2* = {x2_opt:.4f}
U* = {U_opt:.4f}
lambda* = {alpha1 * x2_opt**alpha2 / (x1_opt**(1-alpha1) * p1):.4f}
```

Interpretación económica:

- TMS =  $p_1/p_2: \{(\alpha_1 * x_2)/(\alpha_2 * x_1)\} = \{p_1/p_2\}$
- Gasto total:  $\$ \{p_1 * x_1 + p_2 * x_2\} = \$\{m\}$
- lambda\* = Utilidad marginal del ingreso

"""

```
ax2.text(0.1, 0.95, texto_cpo, fontsize=10, verticalalignment='top',
          family='monospace', bbox=dict(boxstyle='round', facecolor='wheat', alpha=0.3))

plt.tight_layout()
plt.show()

# Verificación numérica de CPO
print("=*60)
print("VERIFICACIÓN NUMÉRICA DE CONDICIONES DE PRIMER ORDEN")
print("=*60)
print(f"\nSolución óptima:")
print(f"  x1* = {x1_opt:.6f}")
print(f"  x2* = {x2_opt:.6f}")
print(f"  U* = {U_opt:.6f}")
print(f"\nVerificación de CPO:")
UMg1 = alpha1 * x1_opt**(alpha1-1) * x2_opt**alpha2
UMg2 = alpha2 * x1_opt**alpha1 * x2_opt**(alpha2-1)
lambda_opt = UMg1 / p1
print(f"  dU/dx1 / p1 = {UMg1/p1:.6f}")
print(f"  dU/dx2 / p2 = {UMg2/p2:.6f}")
print(f"  Igualdad CPO: {abs(UMg1/p1 - UMg2/p2) < 1e-6}")
print(f"\nTMS = p1/p2:")
print(f"  TMS = {((alpha1*x2_opt)/(alpha2*x1_opt)):.6f}")
```

```

print(f" p1/p2 = {p1/p2:.6f}")
print(f" Igualdad: {abs((alpha1*x2_opt)/(alpha2*x1_opt) - p1/p2) < 1e-4}")

```

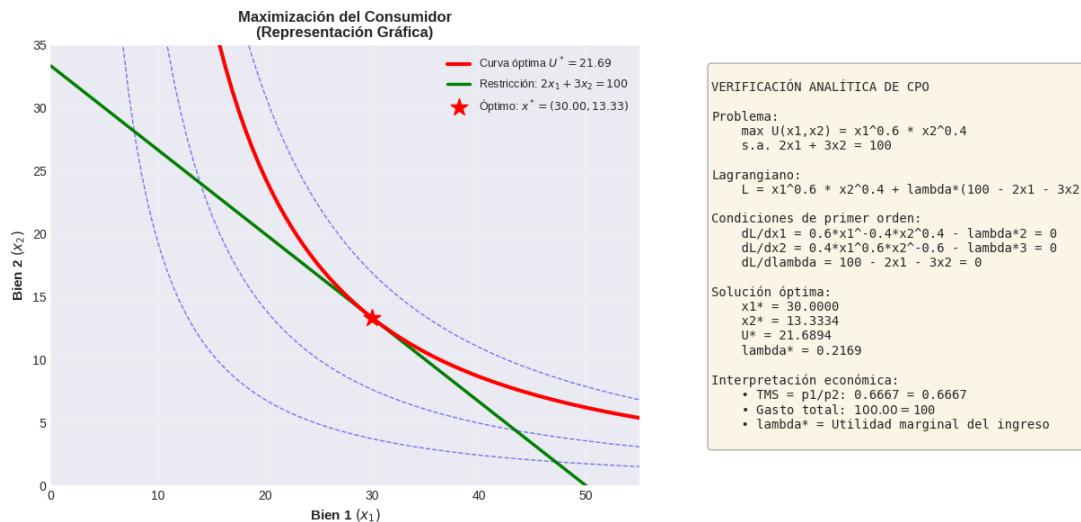


Figura 3: Condiciones de primer orden: maximización del consumidor

---

#### VERIFICACIÓN NUMÉRICA DE CONDICIONES DE PRIMER ORDEN

---

Solución óptima:

$$x_1^* = 29.999959$$

$$x_2^* = 13.333361$$

$$U^* = 21.689435$$

Verificación de CPO:

$$\frac{dU}{dx_1} / p_1 = 0.216895$$

$$\frac{dU}{dx_2} / p_2 = 0.216894$$

Igualdad CPO: True

TMS =  $p_1/p_2$ :

$$TMS = 0.666669$$

$$p_1/p_2 = 0.666667$$

Igualdad: True

### 1.3.3.3 Ventaja 3: Explicitación Completa de Supuestos

Los teoremas matemáticos exigen declaración precisa de **condiciones suficientes** (?):

### ⚠ Teorema 1.2 (Teorema de la Función Implícita - Versión Simplificada)

Sea  $F : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$  una función continuamente diferenciable. Suponga que  $F(x_0, y_0) = 0$  y que la matriz Jacobiana  $\frac{\partial F}{\partial y}$  evaluada en  $(x_0, y_0)$  es no singular. Entonces existen vecindades  $U$  de  $x_0$  y  $V$  de  $y_0$ , y una función única  $g : U \rightarrow V$  tal que:

1.  $g(x_0) = y_0$
2.  $F(x, g(x)) = 0$  para todo  $x \in U$
3.  $g$  es continuamente diferenciable con:

$$\frac{\partial g}{\partial x} = - \left( \frac{\partial F}{\partial y} \right)^{-1} \frac{\partial F}{\partial x}$$

**Importancia económica:** Este teorema garantiza que podemos despejar variables endógenas en función de parámetros exógenos, **siempre que las condiciones sean satisfechas**. La formulación matemática obliga a verificar estas condiciones.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

# Ejemplo: Equilibrio de mercado implícito
# F(P, Q, ) = Qd(P,) - Qs(P) = 0

def F_equilibrio(vars, alpha):
    """Función implícita del equilibrio de mercado"""
    P, Q = vars
    Qd = alpha - 2*P # Demanda
    Qs = -20 + 3*P # Oferta
    return [Q - Qd, Q - Qs]

# Resolver para diferentes valores de
alpha_vals = np.linspace(80, 120, 9)
P_eq = []
Q_eq = []

for alpha in alpha_vals:
    sol = fsolve(lambda vars: F_equilibrio(vars, alpha), [20, 40])
    P_eq.append(sol[0])
    Q_eq.append(sol[1])

P_eq = np.array(P_eq)
Q_eq = np.array(Q_eq)

# Derivada implícita P*/
dP_dalpha_numerical = np.gradient(P_eq, alpha_vals)
dP_dalpha_theoretical = 1/5 # De la solución analítica

fig, axes = plt.subplots(1, 3, figsize=(16, 5))
```

```

# Panel A: Función implícita F=0
ax1 = axes[0]
P_grid = np.linspace(15, 35, 100)
Q_grid = np.linspace(20, 80, 100)
P_mesh, Q_mesh = np.meshgrid(P_grid, Q_grid)

alpha_ejemplo = 100
Qd_mesh = alpha_ejemplo - 2*P_mesh
Qs_mesh = -20 + 3*P_mesh
F_mesh = Qd_mesh - Qs_mesh

contour = ax1.contour(P_mesh, Q_mesh, F_mesh, levels=[0], colors='red', linewidths=3)
ax1.clabel(contour, inline=True, fontsize=10, fmt='F=0')
ax1.scatter(P_eq[4], Q_eq[4], color='red', s=200, marker='*', zorder=5,
            label=f'Equilibrio: P*={P_eq[4]:.1f}, Q*={Q_eq[4]:.1f}')
ax1.set_xlabel('Precio (P)', fontsize=11, fontweight='bold')
ax1.set_ylabel('Cantidad (Q)', fontsize=11, fontweight='bold')
ax1.set_title('Función Implícita $F(P,Q,\alpha)=0$\n(Equilibrio de Mercado)',
              fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend(fontsize=10)

# Panel B: Precio de equilibrio vs parámetro
ax2 = axes[1]
ax2.plot(alpha_vals, P_eq, 'b-o', linewidth=2.5, markersize=8,
          label='$P^*(\alpha)$ (numérico)')
# CORRECCIÓN AQUÍ: Usar comillas dobles para el string interno
ax2.set_xlabel('Parámetro de demanda $(\alpha)$', fontsize=11, fontweight='bold')
ax2.set_ylabel('Precio de equilibrio $P^*$', fontsize=11, fontweight='bold')
ax2.set_title('Función Implícita Despejada\n$P^* = g(\alpha)$',
              fontsize=12, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend(fontsize=10)

# Panel C: Derivada implícita
ax3 = axes[2]
ax3.plot(alpha_vals, dP_dalpha_numerical, 'g-o', linewidth=2.5, markersize=8,
          label='$\frac{dP^*}{d\alpha}$ (numérico)')
ax3.axhline(y=dP_dalpha_theoretical, color='red', linestyle='--', linewidth=2,
            label=f'Teórico: $\frac{{1}}{{5}} = {dP_dalpha_theoretical:.2f}$')
# CORRECCIÓN AQUÍ también
ax3.set_xlabel('Parámetro de demanda $(\alpha)$', fontsize=11, fontweight='bold')
ax3.set_ylabel('Derivada $\frac{dP^*}{d\alpha}$', fontsize=11, fontweight='bold')
ax3.set_title('Derivada Implícita\n(Estática Comparativa)',
              fontsize=12, fontweight='bold')
ax3.grid(True, alpha=0.3)
ax3.legend(fontsize=10)

```

```

plt.tight_layout()
plt.show()

print("=*70")
print("VERIFICACIÓN DEL TEOREMA DE LA FUNCIÓN IMPLÍCITA")
print("=*70")
print("\nSistema de equilibrio:")
print(" Demanda: Qd = - 2P")
print(" Oferta: Qs = -20 + 3P")
print(" Equilibrio: F(P, Q, ) = Qd - Qs = 0")
print("\nSolución analítica:")
print(" P* = ( + 20)/5")
print(" Q* = (3 - 40)/5")
print("\nDerivada implícita (Teorema):")
print(" P*/ = 1/5 = 0.2")
print(f"\nVerificación numérica:")
print(f" Promedio numérico: {np.mean(dP_dalpha_numerical):.4f}")
print(f" Valor teórico: {dP_dalpha_theoretical:.4f}")
print(f" Error relativo: {abs(np.mean(dP_dalpha_numerical) - dP_dalpha_theoretical)/dP_dalpha_theoretical:.4f}")

```

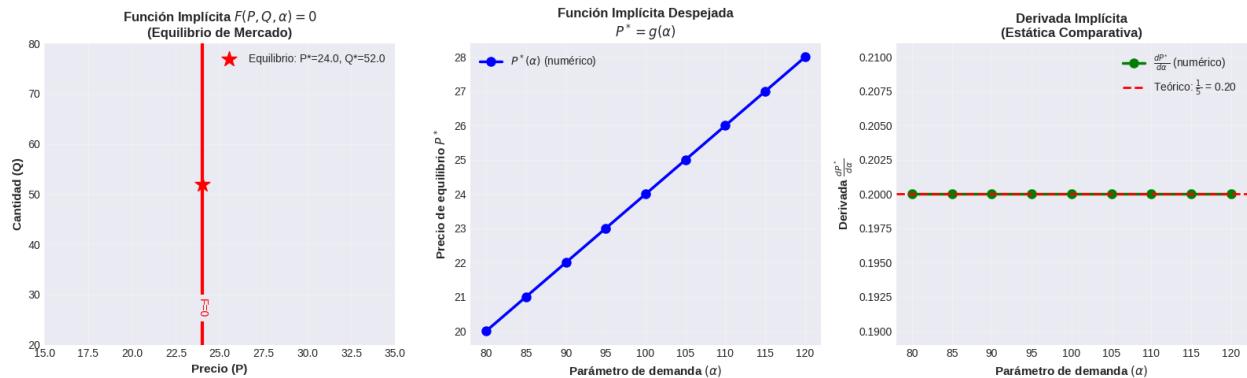


Figura 4: Teorema de la función implícita: ejemplo económico

---

### VERIFICACIÓN DEL TEOREMA DE LA FUNCIÓN IMPLÍCITA

---

Sistema de equilibrio:

Demanda:  $Q_d = -2P$

Oferta:  $Q_s = -20 + 3P$

Equilibrio:  $F(P, Q, \alpha) = Q_d - Q_s = 0$

Solución analítica:

$P^* = ( + 20)/5$

$Q^* = (3 - 40)/5$

Derivada implícita (Teorema):

$$P*/ = 1/5 = 0.2$$

Verificación numérica:

Promedio numérico: 0.2000

Valor teórico: 0.2000

Error relativo: 0.00%

#### 1.3.3.4 Ventaja 4: Tratamiento del Caso General de $n$ Dimensiones

La geometría limita la visualización a 2-3 dimensiones, pero el álgebra maneja  $n$  variables (?):

##### Ejemplo 1.2 (Optimización con $n$ bienes)

```
def resolver_consumidor(n_bienes=10, alpha=None, precios=None, ingreso=100):
    """
        Resuelve el problema del consumidor para n bienes con utilidad Cobb-Douglas

        Parámetros:
        -----
        n_bienes : int
            Número de bienes
        alpha : array
            Exponentes Cobb-Douglas (deben sumar 1)
        precios : array
            Vector de precios
        ingreso : float
            Ingreso del consumidor
    """
    if alpha is None:
        alpha = np.ones(n_bienes) / n_bienes # Pesos iguales
    if precios is None:
        precios = np.random.uniform(1, 5, n_bienes)

    # Solución analítica Cobb-Douglas
    x_optimo = (alpha * ingreso) / precios
    U_optimo = np.prod(x_optimo ** alpha)

    return x_optimo, U_optimo, precios, alpha

# Resolver para diferentes dimensiones
dimensiones = [2, 5, 10, 20, 50, 100]
resultados = []

np.random.seed(42)
for n in dimensiones:
    x_opt, U_opt, p, a = resolver_consumidor(n_bienes=n, ingreso=100)
    gasto_total = np.sum(p * x_opt)
    resultados.append({
```

```

        'n': n,
        'U*': U_opt,
        'gasto': gasto_total,
        'max_x': np.max(x_opt),
        'min_x': np.min(x_opt)
    })

df_resultados = pd.DataFrame(resultados)

# Visualización
fig, axes = plt.subplots(2, 2, figsize=(14, 10))

# Panel A: Utilidad óptima vs dimensión
ax1 = axes[0, 0]
ax1.plot(df_resultados['n'], df_resultados['U*'], 'b-o', linewidth=2.5, markersize=8)
ax1.set_xlabel('Número de bienes (n)', fontsize=11, fontweight='bold')
ax1.set_ylabel('Utilidad óptima $U^*$', fontsize=11, fontweight='bold')
ax1.set_title('Utilidad Óptima vs Dimensión', fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.set_xscale('log')

# Panel B: Verificación de restricción presupuestaria
ax2 = axes[0, 1]
ax2.plot(df_resultados['n'], df_resultados['gasto'], 'r-s', linewidth=2.5, markersize=8)
ax2.axhline(y=100, color='green', linestyle='--', linewidth=2, label='Ingreso = 100')
ax2.set_xlabel('Número de bienes (n)', fontsize=11, fontweight='bold')
ax2.set_ylabel('Gasto total', fontsize=11, fontweight='bold')
ax2.set_title('Verificación: Restricción Presupuestaria', fontsize=12, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend(fontsize=10)
ax2.set_xscale('log')

# Panel C: Ejemplo específico n=10
ax3 = axes[1, 0]
n_ejemplo = 10
x_ej, U_ej, p_ej, a_ej = resolver_consumidor(n_bienes=n_ejemplo, ingreso=100)
bienes = np.arange(1, n_ejemplo+1)
ax3.bar(bienes, x_ej, color='steelblue', alpha=0.7, edgecolor='black')
ax3.set_xlabel('Bien $i$', fontsize=11, fontweight='bold')
ax3.set_ylabel('Cantidad óptima $x_i$', fontsize=11, fontweight='bold')
ax3.set_title(f'Canasta Óptima para n={n_ejemplo} bienes', fontsize=12, fontweight='bold')
ax3.grid(True, alpha=0.3, axis='y')
ax3.set_xticks(bienes)

# Panel D: Tabla resumen
ax4 = axes[1, 1]
ax4.axis('off')

```

```

tabla_texto = df_resultados.to_string(index=False, float_format=lambda x: f'{x:.4f}')
ax4.text(0.1, 0.9, 'RESULTADOS NUMÉRICOS\n' + '='*50 + '\n' + tabla_texto,
         fontsize=9, family='monospace', verticalalignment='top',
         bbox=dict(boxstyle='round', facecolor='lightblue', alpha=0.3))

plt.tight_layout()
plt.show()

print("=*70)
print("OPTIMIZACIÓN EN DIMENSIÓN ARBITRARIA")
print("=*70)
print(f"\nProblema general para n bienes:")
print(r" \max U(x_1, \ldots, x_n) = \prod_i x_i^{\alpha_i}")
print(r" s.a. \sum_i p_i x_i = m")
print(r" x_i^* = (\alpha_i \cdot m) / p_i \quad \forall i \in \{1, \ldots, n\}")
print(f"\nSolución analítica:")
print(r" x_i^* = (\alpha_i \cdot m) / p_i \quad \forall i \in \{1, \ldots, n\}") # ← CAMBIO A
print(f"\nEsta fórmula es válida independientemente de n")
print(f"(la geometría falla para n>3, pero el álgebra no)")

```

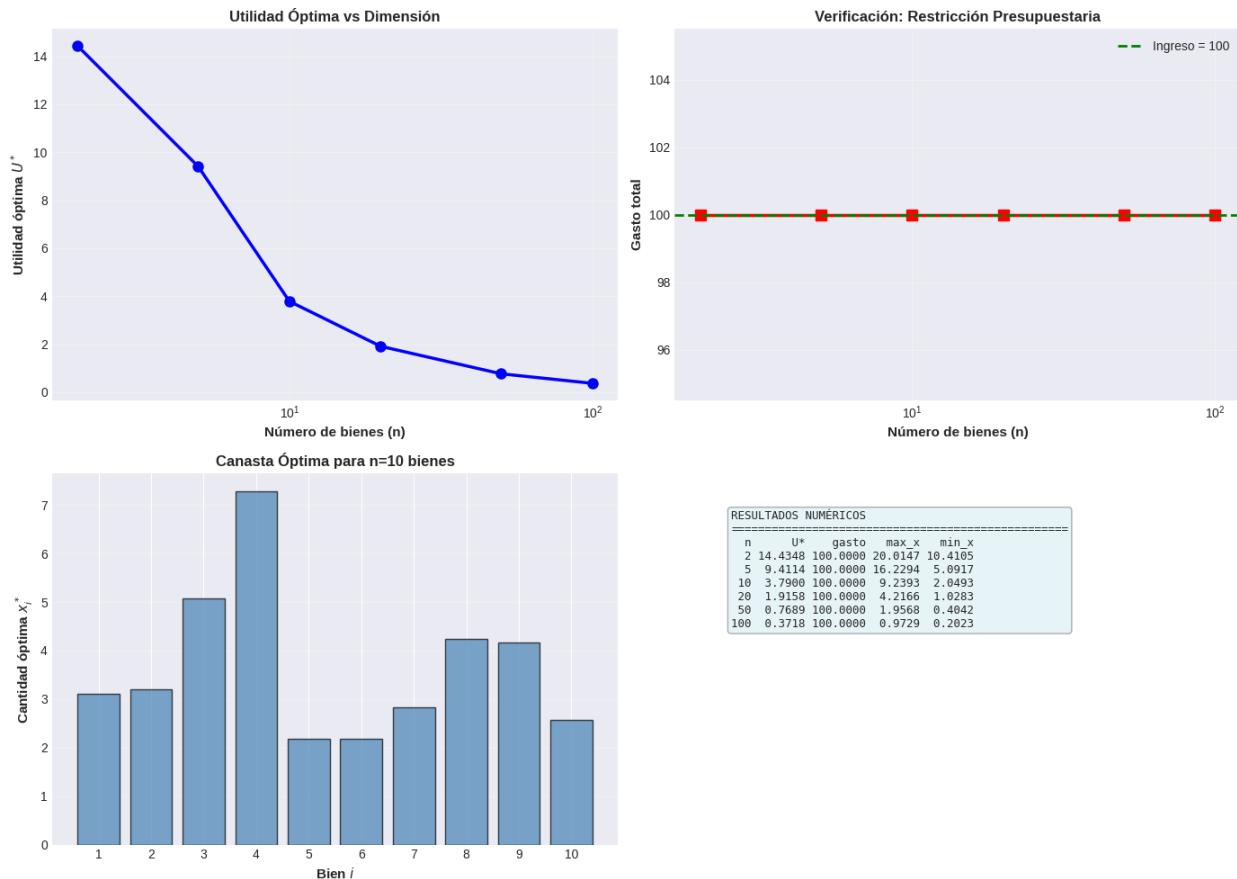


Figura 5: Optimización del consumidor en dimensión arbitraria

---

---

## OPTIMIZACIÓN EN DIMENSIÓN ARBITRARIA

---

---

Problema general para n bienes:

$$\begin{aligned} \max U(x_1, \dots, x_n) &= \prod_i x_i^{\alpha_i} \\ \text{s.a. } \sum_i p_i x_i &= m \\ x_i^* &= (\alpha_i \cdot m) / p_i \quad \forall i \in \{1, \dots, n\} \end{aligned}$$

Solución analítica:

$$x_i^* = (\alpha_i \cdot m) / p_i \quad \forall i \in \{1, \dots, n\}$$

Esta fórmula es válida independientemente de n  
(la geometría falla para n>3, pero el álgebra no)

### 1.3.4 1.2.4 Respuesta a la Crítica de “Irrealidad”

Una crítica común a la economía matemática es que sus modelos son “irreales” o “alejados de la realidad” (?). Esta crítica confunde el **método** con el **contenido**.

! Proposición 1.1 (Defensa del Formalismo)

La crítica de que la teoría matemática es “irreal” es **metodológicamente incorrecta** porque:

1. **Toda teoría científica es abstracta:** La abstracción es necesaria para el análisis científico
2. **El realismo depende de los supuestos, no del método:** Un modelo verbal puede ser tan “irreal” como uno matemático
3. **La matemática es un lenguaje, no un filtro de realidad:** El formalismo no impone restricciones de contenido
4. **La falsabilidad es independiente del formalismo:** Tanto modelos matemáticos como verbales pueden ser empíricamente falsos

**Corolario:** Si un modelo matemático es “irreal”, el problema está en los **supuestos económicos**, no en el **método matemático**.

## 1.4 1.3 Economía Matemática vs. Econometría

### 1.4.1 1.3.1 Distinción Fundamental

La econometría y la economía matemática son disciplinas **complementarias** pero **metodológicamente distintas** (?).

i Definición 1.3 (Econometría)

La **econometría** es la aplicación de métodos matemáticos y estadísticos a datos económicos con los objetivos de:

1. **Estimación:** Cuantificar parámetros de relaciones económicas
2. **Inferencia:** Probar hipótesis sobre comportamientos económicos
3. **Predicción:** Pronosticar valores futuros de variables económicas

**4. Evaluación de políticas:** Medir impactos causales de intervenciones  
 Formalmente, dado un modelo económico  $y = f(x; \theta)$ , la econometría busca estimar  $\hat{\theta}$  a partir de datos observados  $\{(y_i, x_i)\}_{i=1}^N$ .

#### 1.4.2 1.3.2 Comparación Sistemática

Tabla 4: Economía matemática vs. Econometría: comparación detallada

Dimensión	Economía Matemática	Econometría
Objetivo primario	Derivar implicaciones lógicas de supuestos	Estimar relaciones y probar hipótesis
Enfoque metodológico	Deductivo (axiomático)	Inductivo (inferencial)
Naturaleza del análisis	Teórico-cualitativo	Empírico-cuantitativo
Papel de los datos	Abstractos, simbólicos, parametrizados	Observados, muestrales, numéricos
Tipo de conclusiones	Teoremas, proposiciones, condiciones generales	Estimadores, estadísticos, intervalos
Herramientas principales	Cálculo, álgebra lineal, optimización, teoría de juegos	Regresión, máxima verosimilitud, GMM, panel
Preguntas típicas	¿Qué implica X? ¿Bajo qué condiciones Y?	¿Cuánto es X? ¿Es significativo Y?
Validación de resultados	Consistencia lógica, rigor matemático	Significancia estadística, bondad de ajuste

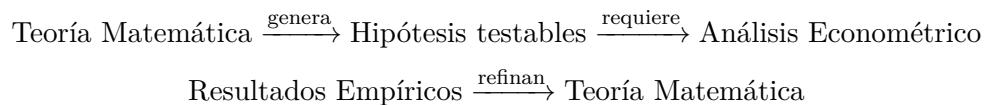
#### 1.4.3 1.3.3 Complementariedad Esencial

##### 💡 Proposición 1.2 (Complementariedad)

La economía matemática y la econometría son **mutuamente necesarias** para el progreso científico en economía:

1. **Teoría sin datos es vacía:** Los modelos matemáticos requieren confrontación empírica
2. **Datos sin teoría son ciegos:** El análisis empírico requiere estructura teórica
3. **Retroalimentación iterativa:** Los resultados empíricos refinan la teoría; la teoría guía el análisis empírico

Esquemáticamente:



#### 1.4.4 1.3.4 Ejemplo Integrado: Curva de Phillips

Ilustremos la complementariedad con un ejemplo clásico (?):

```

# 1. TEORÍA MATEMÁTICA: Modelo de Phillips aumentado por expectativas
#  $\pi_t = \pi_e + (\pi_t - \pi^*) + \epsilon_t$ 

def phillips_teorico(u, u_star, beta, pi_e):
    """Curva de Phillips teórica"""
    return pi_e + beta * (u - u_star)

# Parámetros teóricos
u_star = 5.0 # Tasa natural de desempleo
beta = -0.5 # Sensibilidad de inflación al desempleo
pi_e = 2.0 # Expectativas de inflación

# 2. SIMULACIÓN DE DATOS (representando realidad económica)
np.random.seed(123)
n_obs = 100
u_obs = np.random.uniform(3, 9, n_obs)
pi_teorico = phillips_teorico(u_obs, u_star, beta, pi_e)
# Añadir shocks aleatorios
epsilon = np.random.normal(0, 0.8, n_obs)
pi_obs = pi_teorico + epsilon

# 3. ANÁLISIS ECONOMÉTRICO: Estimación por MCO
from scipy import stats

# Regresión lineal
X = u_obs - u_obs.mean() # Centrar para interpretación
slope, intercept, r_value, p_value, std_err = stats.linregress(u_obs, pi_obs)

print("=*70")
print("ANÁLISIS INTEGRADO: CURVA DE PHILLIPS")
print("=*70")
print("\n1. TEORÍA MATEMÁTICA:")
print(r"  Modelo:  $\pi = \pi^e + \beta(u - u^*) + \epsilon$ ")
print(rf"  - Coeficiente  $\beta$ : {beta:.2f}")
print(rf"  - Expectativas  $\pi^e$ : {pi_e:.1f}%")
print(rf"  - Intercepto: {intercept:.3f} ( $\pi^e$  estimado)")
print(rf"  - Pendiente: {slope:.3f} ( $\beta$  estimado)")

print("\n2. ANÁLISIS ECONOMÉTRICO:")
print(f"  Estimación por MCO:")
print(f"  - Intercepto: {intercept:.3f} (^e estimado)")
print(f"  - Pendiente: {slope:.3f} (\beta estimado)")
print(f"  - R: {r_value**2:.3f}")
print(f"  - p-value: {p_value:.6f}")
print(f"  - Error estándar: {std_err:.3f}")

print("\n3. COMPARACIÓN TEORÍA-EVIDENCIA:")

```

```

print(f" \beta teórico vs estimado: {beta:.3f} vs {slope:.3f}")
print(f" Error de estimación: {abs(beta - slope):.3f}")
print(f" Error relativo: {abs(beta - slope)/abs(beta)*100:.1f}%")

# Visualización integrada
fig, axes = plt.subplots(2, 2, figsize=(15, 12))

# Panel A: Teoría pura
ax1 = axes[0, 0]
u_range = np.linspace(0, 10, 100)
pi_teoria = phillips_teorico(u_range, u_star, beta, pi_e)
ax1.plot(u_range, pi_teoria, 'b-', linewidth=3, label='Teoría: $\pi = 2 - 0.5(u - 5)$')
ax1.axvline(x=u_star, color='red', linestyle='--', alpha=0.7, label=f'$u^* = {u_star}$')
ax1.axhline(y=pi_e, color='green', linestyle='--', alpha=0.7, label=f'$\pi^e = {pi_e}$')
ax1.set_xlabel('Tasa de desempleo (u) %', fontsize=11, fontweight='bold')
ax1.set_ylabel('Tasa de inflación () %', fontsize=11, fontweight='bold')
ax1.set_title('A. Modelo Teórico\n(Economía Matemática)', fontsize=12, fontweight='bold')
ax1.grid(True, alpha=0.3)
ax1.legend(fontsize=10)
ax1.set_xlim(0, 10)
ax1.set_ylim(-2, 6)

# Panel B: Datos observados
ax2 = axes[0, 1]
ax2.scatter(u_obs, pi_obs, alpha=0.5, s=50, color='red', label='Datos observados')
ax2.plot(u_obs, intercept + slope * u_obs, 'g--', linewidth=2.5,
         label=f'MCO: = {intercept:.2f} + {slope:.2f}u')
ax2.set_xlabel('Tasa de desempleo (u) %', fontsize=11, fontweight='bold')
ax2.set_ylabel('Tasa de inflación () %', fontsize=11, fontweight='bold')
ax2.set_title('B. Análisis Empírico\n(Econometría)', fontsize=12, fontweight='bold')
ax2.grid(True, alpha=0.3)
ax2.legend(fontsize=10)
ax2.set_xlim(0, 10)
ax2.set_ylim(-2, 6)

# Panel C: Integración
ax3 = axes[1, 0]
ax3.plot(u_range, pi_teoria, 'b-', linewidth=3, label='Teoría', alpha=0.7)
ax3.scatter(u_obs, pi_obs, alpha=0.3, s=30, color='red', label='Datos')
ax3.plot(u_obs, intercept + slope * u_obs, 'g--', linewidth=2.5, label='Estimación')
ax3.set_xlabel('Tasa de desempleo (u) %', fontsize=11, fontweight='bold')
ax3.set_ylabel('Tasa de inflación () %', fontsize=11, fontweight='bold')
ax3.set_title('C. Integración Teoría-Evidencia', fontsize=12, fontweight='bold')
ax3.grid(True, alpha=0.3)
ax3.legend(fontsize=10)
ax3.set_xlim(0, 10)
ax3.set_ylim(-2, 6)

```

```

# Panel D: Residuos (diagnóstico)
ax4 = axes[1, 1]
residuos = pi_obs - (intercept + slope * u_obs)
ax4.scatter(u_obs, residuos, alpha=0.5, s=50, color='purple')
ax4.axhline(y=0, color='black', linestyle='-', linewidth=1)
ax4.axhline(y=2*np.std(residuos), color='red', linestyle='--', alpha=0.5)
ax4.axhline(y=-2*np.std(residuos), color='red', linestyle='--', alpha=0.5)
ax4.set_xlabel('Tasa de desempleo (u) %', fontsize=11, fontweight='bold')
ax4.set_ylabel('Residuos', fontsize=11, fontweight='bold')
ax4.set_title('D. Diagnóstico de Residuos\n(Validación del Modelo)', fontsize=12, fontweight='bold')
ax4.grid(True, alpha=0.3)
ax4.set_xlim(0, 10)

plt.tight_layout()
plt.show()

# Test de normalidad de residuos
from scipy.stats import shapiro
stat_shapiro, p_shapiro = shapiro(residuos)
print("\n4. DIAGNÓSTICO ESTADÍSTICO:")
print(f"    Test de Shapiro-Wilk (normalidad residuos):")
print(f"        - Estadístico: {stat_shapiro:.4f}")
print(f"        - p-value: {p_shapiro:.4f}")
print(f"        - Conclusión: {'Residuos normales' if p_shapiro > 0.05 else 'Evidencia contra norma'}
```

=====
ANÁLISIS INTEGRADO: CURVA DE PHILLIPS
=====

1. TEORÍA MATEMÁTICA:

- Modelo:  $\hat{\pi} = \beta_0 + \beta_1(u - u^*)$
- Coeficiente  $\beta_1$ : -0.50
  - Expectativas  $\hat{\pi}^e$ : 2.0%
  - Intercepto: 4.744 ( $\hat{\pi}^e$  estimado)
  - Pendiente: -0.536 ( $\beta_1$  estimado)

2. ANÁLISIS ECONOMÉTRICO:

Estimación por MCO:

- Intercepto: 4.744 ( $\hat{\pi}^e$  estimado)
- Pendiente: -0.536 ( $\beta_1$  estimado)
- R: 0.504
- p-value: 0.000000
- Error estándar: 0.054

3. COMPARACIÓN TEORÍA-EVIDENCIA:

$\hat{\pi}$  teórico vs estimado: -0.500 vs -0.536

Error de estimación: 0.036

Error relativo: 7.2%

#### 4. DIAGNÓSTICO ESTADÍSTICO:

Test de Shapiro-Wilk (normalidad residuos):

- Estadístico: 0.9939
- p-value: 0.9340
- Conclusión: Residuos normales

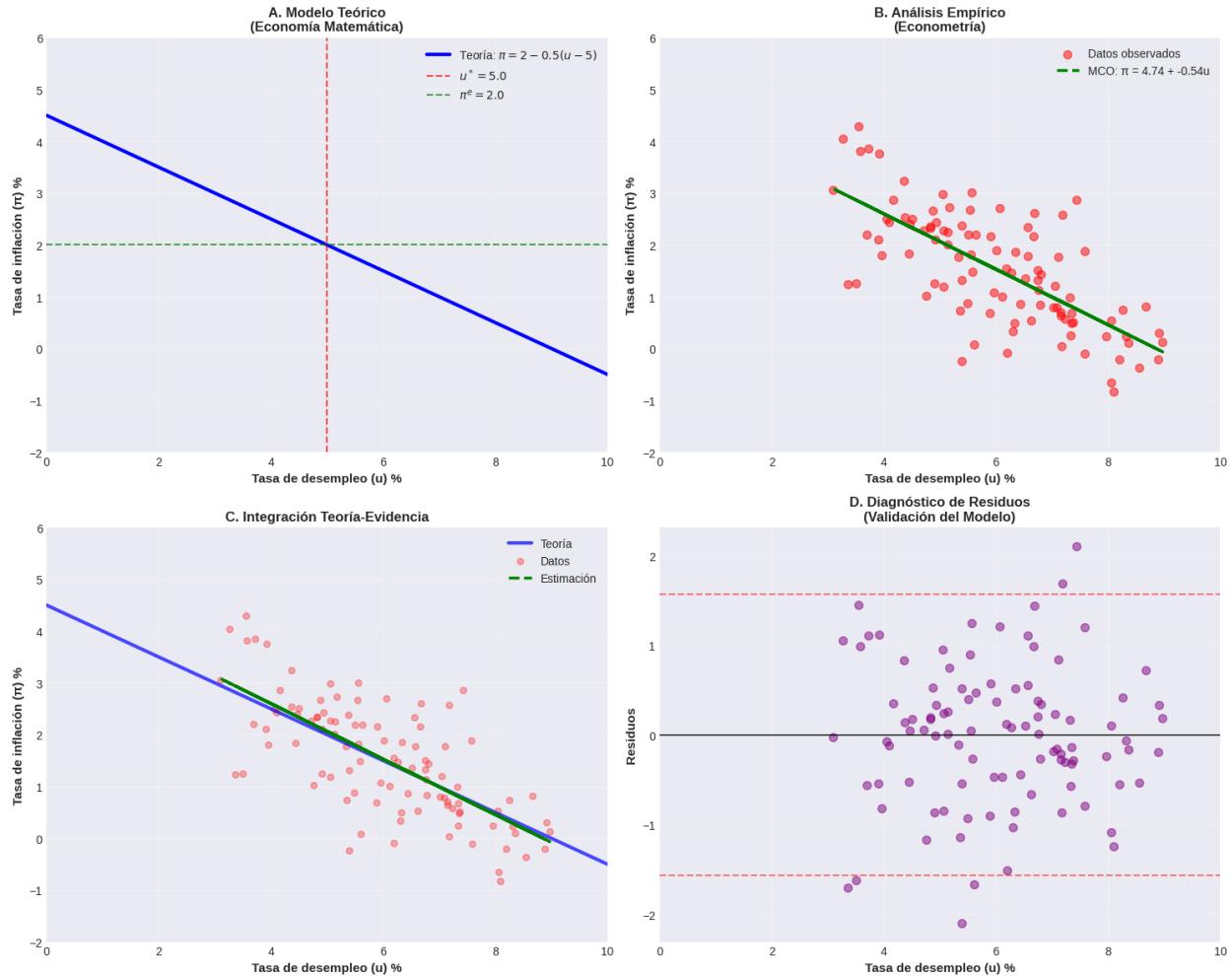


Figura 6: Curva de Phillips: integración de teoría matemática y análisis empírico

### 1.5 1.4 Aplicaciones Avanzadas

#### 1.5.1 1.4.1 Optimización del Productor con Retornos Constantes a Escala

Desarrollaremos completamente un problema de optimización del productor (?).

**Problema:** Una firma con tecnología Cobb-Douglas busca minimizar costos:

**Problema:** Una firma con tecnología Cobb-Douglas busca minimizar costos:

$$\min_{K,L} rK + wL \quad \text{sujeto a} \quad AK^\alpha L^{1-\alpha} \geq Q_0$$

donde  $r$  es el precio del capital,  $w$  el salario, y  $0 < \alpha < 1$ .

## 1.6 Referencias

- Chiang, A. C., & Wainwright, K. (2005). *Fundamental Methods of Mathematical Economics* (4th ed.). McGraw-Hill.
- Debreu, G. (1959). *Theory of Value: An Axiomatic Analysis of Economic Equilibrium*. Yale University Press.
- Simon, C. P., & Blume, L. (1994). *Mathematics for Economists*. W.W. Norton & Company.
- Varian, H. R. (1992). *Microeconomic Analysis* (3rd ed.). W.W. Norton & Company.

---

**Emanuel Quintana Silva**

Universidad Pedagógica y Tecnológica de Colombia (UPTC)