Module 6 — Keynes and the Multiplier

The Great Depression, Aggregate Demand, and Fiscal Policy

Dr Juan Zurita

From Smith to Simulation

The University of Edinburgh · School of Economics

Part I — The History

October 1929: Black Tuesday

- October 29, 1929 — the New York Stock Exchange crashes. Billions of dollars in wealth evaporate within weeks.
- What followed was the **Great Depression**: the most severe economic catastrophe of the modern era.
- US GDP fell by nearly 30% (1929–1933).
- Unemployment rose to 25%.
- World trade collapsed by two-thirds.

The Scale of Disaster

Factories stood idle. Millions who wanted to work could not find jobs. And the prevailing economic orthodoxy had no good explanation for *why* — and no good prescription for what to do.

The Classical Orthodoxy: Say's Law

Say's Law (1803)

"Supply creates its own demand."

- A general glut — an economy-wide shortage of demand — was deemed *impossible*.
- If people saved more, interest rates would fall, investment would rise, equilibrium would be restored.
- The prescription: **patience**. The market will sort itself out.

But as the Depression dragged on year after year, patience began to look less like wisdom and more like indifference.

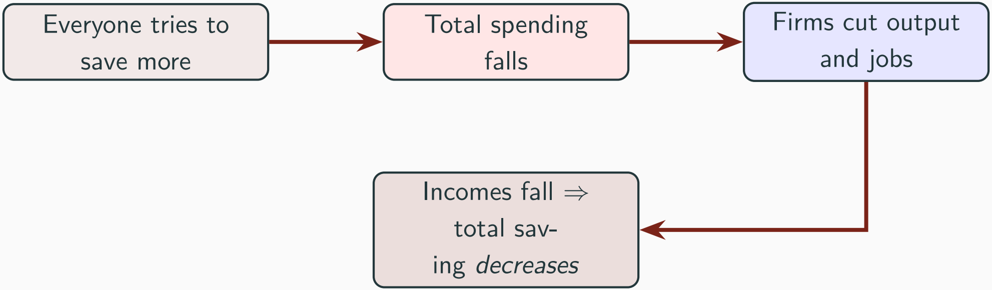
John Maynard Keynes (1883–1946)

- Cambridge economist, investor, patron of the arts, and one of the most brilliant minds of the 20th century.
- In **1936**, he published *The General Theory of Employment, Interest, and Money*.
- Central argument: demand matters.
- An economy can get stuck where people want to work but firms will not hire, because firms do not expect enough demand.
- This is not a temporary glitch — it can persist **indefinitely**.

"The outstanding faults of the economic society in which we live are its failure to provide for full employment and its arbitrary and inequitable distribution of wealth and incomes."

— Keynes, *General Theory* (1936), Ch. 24

Key Idea 1: The Paradox of Thrift



What is rational for each individual is disastrous for the economy as a whole.

Smith's invisible hand, which harmonises self-interest and social welfare in the market for oats, **breaks down** at the level of the whole economy.

Key Idea 2: Animal Spirits

- Investment depends not just on interest rates (as the classics assumed) but on business confidence — what Keynes called "animal spirits".
- When entrepreneurs are optimistic, they invest. When they are pessimistic, they do not.
- Pessimism can be **self-fulfilling**: if firms expect a recession, they cut investment, which *causes* the recession.

A Vicious Circle

Pessimism $\Rightarrow$ less investment $\Rightarrow$ less output $\Rightarrow$ less income $\Rightarrow$ more pessimism $\Rightarrow \ldots$

Key Idea 3: The Multiplier

When the government spends an extra pound:

1. That pound becomes someone's income.
2. They spend part of it — that becomes someone *else's* income.
3. They spend part of *that* — and so on.

The total effect on GDP is a **multiple** of the original spending.

$$\text{Multiplier} = \frac{1}{1 - c_1}$$

where c_1 is the marginal propensity to consume (MPC) — the fraction of each extra pound of income that gets spent.

With $c_1 = 0.8$: multiplier $= 5$. Every £1 of government spending generates £5 of output.

The Edinburgh Connection

Keynesian Debates at Edinburgh

Keynes's ideas were fiercely debated in Edinburgh. The University's economics department in the 1930s and 40s included several economists who were **sceptical** of Keynes — they belonged more to the classical tradition that Keynes was attacking.

This tension between Keynesian macroeconomics and classical market-clearing models remains alive today, and Edinburgh's department has contributed to **both sides** of the argument.

A university where intellectual disagreement is taken seriously
is a university doing its job.

Part II — The Computation

Setting Up

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

- `numpy` — numerical arrays and operations
- `matplotlib` — plotting
- `fsolve` — numerical root-finder (finds equilibrium by solving $f(x) = 0$)

The Keynesian Cross

The simplest Keynesian model has three sources of spending:

$$
\begin{aligned}
C &= c_0 + c_1(Y - T) && \text{(Consumption)} \\
I &= \bar{I} && \text{(Investment — fixed by animal spirits)} \\
G &= \bar{G} && \text{(Government spending — policy choice)}
\end{aligned}
$$

c_0 Autonomous consumption (spending even if income is zero).

c_1 **MPC**: fraction of each extra pound of disposable income spent.

$Y - T$ Disposable income (income minus taxes).

Equilibrium: output = total spending:

$$Y = C + I + G = c_0 + c_1(Y - T) + \bar{I} + G$$

Defining the Model in Python

```python
# Parameters
c0 = 100     # autonomous consumption
c1 = 0.6     # marginal propensity to consume (MPC)
I_bar = 200 # investment (animal spirits)
G = 150      # government spending
T = 100      # taxes

def consumption(Y, c0=c0, c1=c1, T=T):
    return c0 + c1 * (Y - T)

def total_spending(Y, c0=c0, c1=c1, I_bar=I_bar, G=G, T=T):
    return consumption(Y, c0, c1, T) + I_bar + G

def equilibrium_condition(Y, c0=c0, c1=c1, I_bar=I_bar, G=G, T=T):
    """Returns zero when Y = C + I + G."""
    return Y - total_spending(Y, c0, c1, I_bar, G, T)
```

Finding Equilibrium with `fsolve`

Analytical (for the linear model):

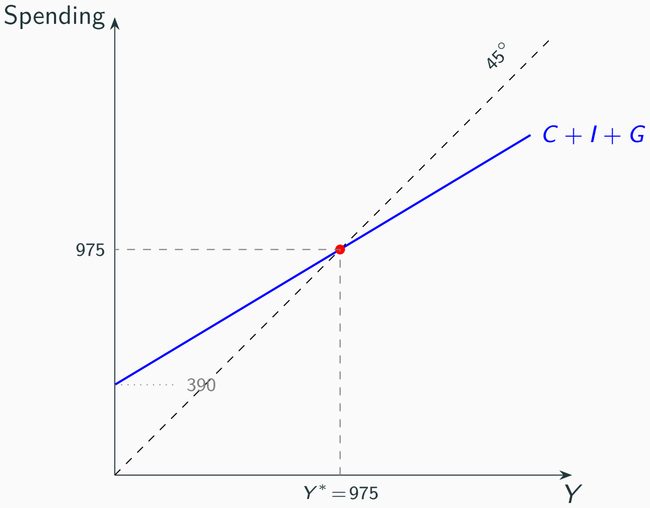
$$Y^* = \frac{c_0 - c_1 T + \bar{I} + G}{1 - c_1} = \frac{100 - 60 + 200 + 150}{0.4} = \frac{390}{0.4} = 975$$

Numerical (works for *any* specification):

```
Y_star = fsolve(equilibrium_condition, x0=500)[0]
C_star = consumption(Y_star)

print(f"Y* = {Y_star:.2f}")    # 975.00
print(f"C* = {C_star:.2f}")    # 625.00
print(f"C + I + G = {C_star + I_bar + G:.2f}")  # 975.00
```

Numerical and analytical solutions agree to machine precision.

Visualising the Keynesian Cross



Equilibrium is where the spending line crosses the 45° line: output equals planned spending.

The Fiscal Multiplier

What happens when the government increases spending by ΔG?

$$Y^* = \frac{c_0 - c_1 T + \bar{I} + G}{1 - c_1} \quad \Longrightarrow \quad \frac{\Delta Y}{\Delta G} = \frac{1}{1 - c_1}$$

With $c_1 = 0.6$

$$\text{Multiplier} = \frac{1}{1 - 0.6} = 2.5$$

Every £1 spent $\Rightarrow$ £2.50 of output.

With $c_1 = 0.8$

$$\text{Multiplier} = \frac{1}{1 - 0.8} = 5.0$$

Every £1 spent $\Rightarrow$ £5.00 of output.

The MPC c_1 is the **engine** of the multiplier. Higher MPC $\Rightarrow$ larger multiplier $\Rightarrow$ more potent fiscal policy.

Computing the Multiplier

```
delta_G = 50
G_new = G + delta_G

Y_star_new = fsolve(equilibrium_condition, x0=500,
                    args=(c0, c1, I_bar, G_new, T))[0]

delta_Y = Y_star_new - Y_star
multiplier_numerical = delta_Y / delta_G
multiplier_theory = 1 / (1 - c1)

print(f"Original Y*: {Y_star:.2f}")      # 975.00
print(f"New Y*:      {Y_star_new:.2f}")   # 1100.00
print(f"Delta Y:     {delta_Y:.2f}")      # 125.00
print(f"Numerical multiplier:   {multiplier_numerical:.4f}")
print(f"Theoretical multiplier: {multiplier_theory:.4f}")
# Both equal 2.5000
```

The Multiplier Round by Round

Trace how a $\Delta G = 50$ ripples through the economy:

Round	Who earns?	New spending	Cumulative
1	Road builders	50.00	50.00
2	Shopkeepers	30.00	80.00
3	Suppliers	18.00	98.00
4	Their suppliers	10.80	108.80
5	...	6.48	115.28
∞		0	**125.00**

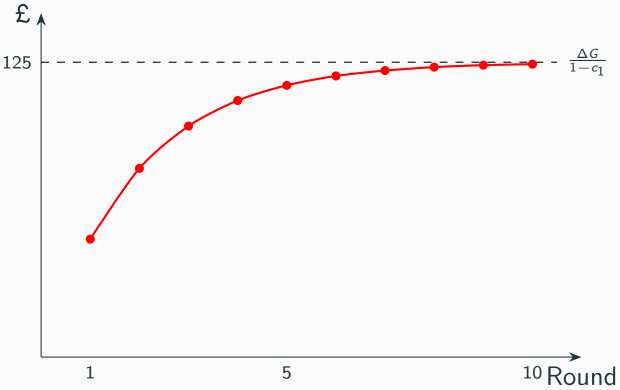
$$\Delta G + \Delta G \cdot c_1 + \Delta G \cdot c_1^2 + \cdots = \frac{\Delta G}{1 - c_1} = \frac{50}{0.4} = 125$$

A geometric series — each round smaller, but they **add up**.

Visualising the Multiplier Process

```
n_rounds = 20
spending_per_round = [delta_G * c1**r for r in range(n_rounds)]
cumulative = np.cumsum(spending_per_round)

fig, axes = plt.subplots(1, 2, figsize=(12, 4.5))
axes[0].bar(range(1, n_rounds+1), spending_per_round,
            color='steelblue', alpha=0.7)
axes[0].set_xlabel('Round'); axes[0].set_ylabel('New spending')
axes[0].set_title('Spending per Round')

axes[1].plot(range(1, n_rounds+1), cumulative, 'o-', color='coral')
axes[1].axhline(delta_G/(1-c1), color='k', linestyle='--',
                label=f'Limit = {delta_G/(1-c1):.0f}')
axes[1].set_xlabel('Round')
axes[1].set_title('Cumulative Multiplier Effect')
axes[1].legend()
plt.tight_layout(); plt.show()
```

Cumulative Multiplier: Convergence



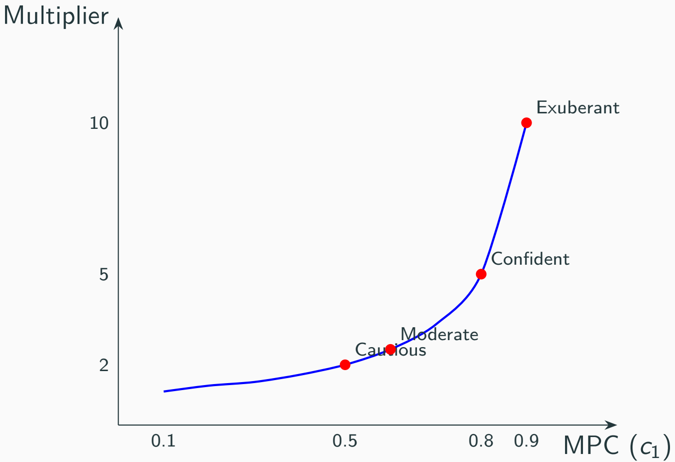
The cumulative effect converges quickly to the theoretical multiplier. After 10 rounds, we have captured 98% of the total effect.

How the MPC Changes the Multiplier

```
c1_values = np.linspace(0.1, 0.95, 200)
multipliers = 1 / (1 - c1_values)

fig, ax = plt.subplots(figsize=(8, 5))
ax.plot(c1_values, multipliers, color='steelblue', linewidth=2.5)

for c1_mark, name in [(0.5, 'Cautious'), (0.6, 'Moderate'),
                       (0.8, 'Confident'), (0.9, 'Exuberant')]:
    m = 1 / (1 - c1_mark)
    ax.plot(c1_mark, m, 'o', markersize=8)
    ax.annotate(f'{name}\nc1={c1_mark}, m={m:.1f}',
                xy=(c1_mark, m), xytext=(c1_mark-0.08, m+1.5),
                fontsize=9, ha='center')
ax.set_xlabel('MPC'); ax.set_ylabel('Multiplier = 1/(1-c1)')
ax.set_title('The Multiplier Depends on How Much People Spend')
plt.show()
```

As $c_1 \to 1$ the multiplier explodes; as $c_1 \to 0$ it shrinks to 1.

MPC Sensitivity: Intuition



"The MPC is the crucial behavioural parameter." — Keynes.

Economies where people spend more of each extra pound see **larger multiplier effects** from

The IS Curve: Adding Interest Rates

In the real economy, investment responds to the interest rate r:

$$I(r) = \bar{I} - d \cdot r$$

where d measures investment sensitivity to the interest rate.

Equilibrium now depends on r. For each interest rate, there is a different equilibrium output Y^*.

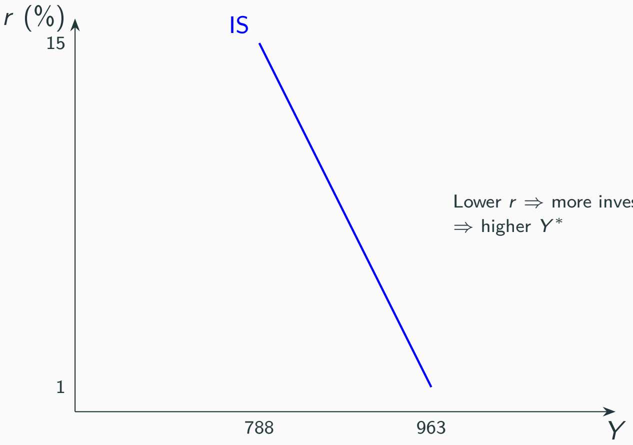
$$Y = c_0 + c_1(Y - T) + (\bar{I} - d\, r) + G$$

The set of all (Y, r) pairs where the goods market is in equilibrium is the IS curve ("Investment = Saving").

Computing the IS Curve

```python
d = 500  # investment sensitivity to interest rate

def investment(r, I_bar=200, d=500):
    return I_bar - d * r

def is_equation(Y, r, c0=100, c1=0.6, I_bar=200, d=500,
                G=150, T=100):
    C = c0 + c1 * (Y - T)
    I = I_bar - d * r
    return Y - C - I - G

# For each r, find equilibrium Y
r_values = np.linspace(0.01, 0.15, 50)
Y_is = np.array([fsolve(is_equation, x0=500, args=(r,))[0]
                  for r in r_values])

fig, ax = plt.subplots(figsize=(8, 5))
ax.plot(Y_is, r_values * 100, color='steelblue', linewidth=2.5)
ax.set_xlabel('Output (Y)'); ax.set_ylabel('Interest rate (%)')
ax.set_title('The IS Curve')
ax.set_ylim(16, 0)
plt.show()
```

The IS Curve: Graphical View



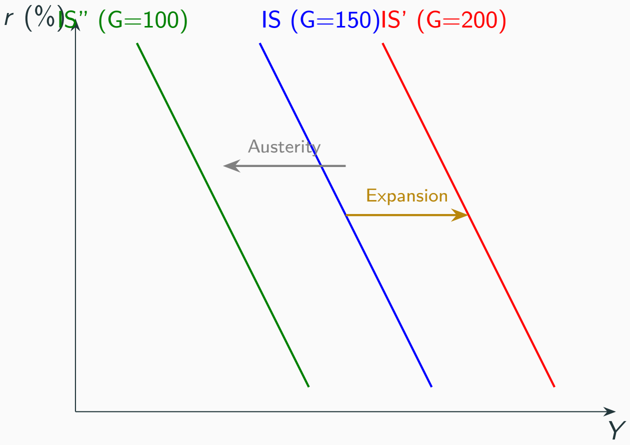
The IS curve slopes **downward**: lower $r \Rightarrow$ cheaper borrowing $\Rightarrow$ more investment $\Rightarrow$ higher equilibrium output.

Fiscal Policy Shifts the IS Curve

```
fig, ax = plt.subplots(figsize=(8, 5))

for G_val, color, label in [
        (150, 'steelblue', 'G = 150 (original)'),
        (200, 'coral',     'G = 200 (expansion)'),
        (100, 'seagreen',  'G = 100 (austerity)')]:
    Y_is_shift = np.array([
        fsolve(is_equation, x0=500,
               args=(r, c0, c1, I_bar, d, G_val, T))[0]
        for r in r_values])
    ax.plot(Y_is_shift, r_values*100, color=color,
            linewidth=2, label=label)

ax.set_xlabel('Output (Y)'); ax.set_ylabel('Interest rate (%)')
ax.set_title('Fiscal Policy Shifts the IS Curve')
ax.legend(); ax.set_ylim(16, 0)
plt.show()
```

Keynes's prescription: shift the IS curve **right** by increasing G.

IS Shifts: The Graphical Argument

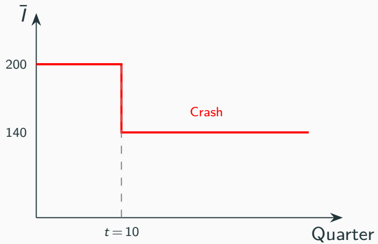


At every interest rate, higher G means higher equilibrium output.
The government spends what the private sector will not.

Animal Spirits: Confidence Crash Simulation

Simulate 40 quarters:

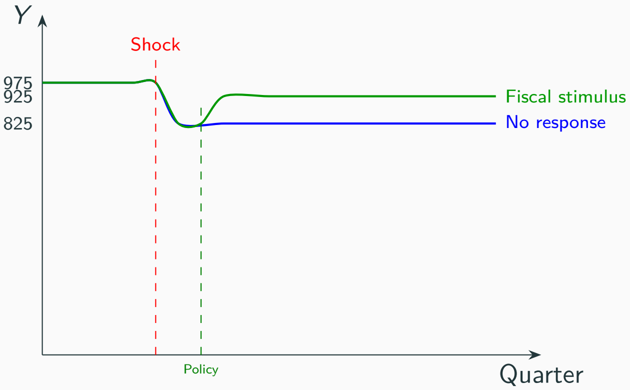
- Quarters 1–10: $\bar{I} = 200$ (normal confidence).
- Quarter 10: confidence crashes $\Rightarrow \bar{I} = 140$.
- **Scenario A:** No policy response.
- **Scenario B:** Government raises G by 40 two quarters later (quarter 12).



Animal Spirits: Python Simulation

```python
T_sim = 40; r_fixed = 0.05
I_bar_path = np.ones(T_sim) * 200
I_bar_path[10:] = 140  # confidence crash at quarter 10

Y_path = np.array([
    fsolve(is_equation, 500,
           args=(r_fixed, c0, c1, I_bar_path[t], d, G, T))[0]
    for t in range(T_sim)])

# With fiscal response: G increases by 40 at quarter 12
G_resp = np.ones(T_sim) * G
G_resp[12:] = G + 40

Y_path_resp = np.array([
    fsolve(is_equation, 500,
           args=(r_fixed, c0, c1, I_bar_path[t], d, G_resp[t], T))[0]
    for t in range(T_sim)])

print(f"Drop without intervention: {Y_path[0]-Y_path[15]:.0f}")
print(f"Drop with fiscal stimulus: {Y_path[0]-Y_path_resp[15]:.0f}")
```

Keynes vs. Doing Nothing



Government spending **partially offsets** the confidence crash.

"When animal spirits fail, the government must step in to maintain demand." — Keynes's core policy argument.

Part III — Exercises & Discussion

Exercise 1: The Paradox of Thrift

Starting from the baseline ($c_0 = 100,\ c_1 = 0.6,\ \bar{I} = 200,\ G = 150,\ T = 100$):

1. Compute equilibrium Y^* for $c_1 = 0.6,\ 0.5,\ 0.4,\ 0.3$.
   Show that output *falls* as people save more.
2. Compute total saving $S = Y - C - T$ at each equilibrium.
   Does saving actually *increase* when people try to save more?
3. Plot Y^* and S against c_1. This is the paradox of thrift in action.
4. In 2–3 sentences: why is what is rational for each individual harmful for the economy as a whole? How does this relate to Smith's invisible hand?

Exercise 2: The Balanced-Budget Multiplier

A politician proposes increasing G by £100, financed entirely by a £100 tax increase ($\Delta G = \Delta T = 100$).

1. Compute Y^* with $G = 150$, $T = 100$ (baseline).
2. Compute Y^* with $G = 250$, $T = 200$ (balanced-budget expansion).
3. What is ΔY? Is the balanced-budget multiplier equal to 0, 1, or something else?
4. Explain intuitively why a balanced-budget expansion still increases output.

Hint

The government spends 100% of the extra revenue, but taxpayers only reduce their spending by $c_1 \times 100$. The net injection is $(1 - c_1) \times 100$.

Exercise 3: Progressive Tax as Automatic Stabiliser

Replace the lump-sum tax T with a **proportional income tax**: $T(Y) = \tau \cdot Y$.

The consumption function becomes:

$$C = c_0 + c_1\,(1 - \tau)\,Y$$

1. Write a new equilibrium condition. Use $\tau = 0.25$.
2. Find Y^* with `fsolve`. Compare to the fixed-tax model.
3. Derive the new multiplier formula:

   $$\text{Multiplier} = \frac{1}{1 - c_1(1 - \tau)}$$

   Verify numerically by increasing G by 50.
4. Compare multipliers for $\tau = 0,\ 0.25,\ 0.5$.
   Keynes's followers called the income tax an automatic stabiliser — explain why.

Key Takeaways

What We Learned This Week

1. The **Great Depression** (1929–33) shattered the classical belief that markets always self-correct.
2. Keynes (1936) argued that aggregate demand can be persistently deficient — economies can get stuck below full employment.
3. The **consumption function** $C = c_0 + c_1(Y - T)$ and the **Keynesian cross** provide a tractable model of output determination.
4. The **fiscal multiplier** $1/(1 - c_1)$ shows that government spending has amplified effects on output.
5. The **IS curve** links interest rates to equilibrium output and illustrates how fiscal policy shifts aggregate demand.
6. **Animal spirits** can cause self-fulfilling recessions; fiscal policy can partially offset confidence collapses.

Further Reading

- Keynes, J.M. *The General Theory of Employment, Interest, and Money* (1936), Chs. 1–3, 10 (the multiplier).
- Skidelsky, R. *Keynes: A Very Short Introduction* — an excellent 150-page overview.
- Heilbroner, R. *The Worldly Philosophers*, Ch. 9 ("The Heresies of John Maynard Keynes").
- Hicks, J.R. "Mr. Keynes and the 'Classics' " (1937) — the paper that invented the IS-LM model.

Next week: Marx and the Machinery Question —
capital accumulation, the labour share of income,
and the debate over whether technology helps or hurts workers.