STEP-BY-STEP EXCEL FINANCIAL MODELLING PROCESS

Excel Financial Modelling Algorithm

Step 0: Initialize Workbook

1. Open Excel → Create new workbook.
2. Add worksheets:
    - Assumptions
    - Income Statement (P&L)
    - Balance Sheet (BS)
    - Cash Flow Statement (CF)
    - Valuation
    - Scenario/Sensitivity

Step 1: Assumptions Sheet

1. List key drivers:
    - Revenue growth rate, gross margin, operating expenses
    - Working capital days: Receivables, Payables, Inventory
    - CapEx, financing assumptions, tax rate, WACC
2. Assign **named ranges** to inputs:
    - Example: Select revenue growth cell → Name = Revenue_Growth
3. Color-code inputs (Blue for assumptions).

Step 2: Timeline Setup

1. Choose projection frequency (monthly, quarterly, or yearly).
2. In all sheets, create timeline columns:
    - Example: B1 = 2025, C1 = 2026, …
3. Use formulas for rolling dates:
    - Monthly: =EDATE(B1,1)
    - Yearly: =B1+1

Step 3: Income Statement (P&L)

1. Revenue:
2. Revenue_Current = Revenue_Previous * (1 + Revenue_Growth)
3. COGS = Revenue * COGS_Percentage
4. Gross Profit = Revenue - COGS
5. Operating Expenses = link to Assumptions
6. EBIT = Gross Profit - Operating Expenses
7. Interest Expense = based on debt schedule
8. Taxes = EBIT * Tax_Rate
9. Net Income = EBIT - Interest - Taxes
10. Link all formulas to Assumptions (no hardcoding).

Step 4: Balance Sheet

1. **Assets**:
    - Cash = link to CF Ending Cash
    - Accounts Receivable = Revenue * Receivable_Days / 365
    - Inventory = COGS * Inventory_Days / 365
    - PP&E = Previous_PP&E + CapEx - Depreciation
2. **Liabilities**:
    - Accounts Payable = COGS * Payable_Days / 365
    - Debt = link to Assumptions / Debt Schedule
3. **Equity** = Previous Equity + Net Income - Dividends
4. **Check**: Assets = Liabilities + Equity

Step 5: Cash Flow Statement

1. Start with Net Income
2. Add non-cash adjustments: Depreciation, Amortization
3. Adjust for changes in working capital:
4. ΔWC = ΔReceivables + ΔInventory - ΔPayables
5. Subtract CapEx

6. Add / Subtract financing flows (Debt, Dividends)
7. Ending Cash = Opening Cash + Net Cash Flow → link to Balance Sheet

Step 6: Implement Financial Formulas

1. NPV(rate, FCF_Range) → Discounted cash flows
2. XNPV(rate, FCF_Range, Dates) → Discounted cash flows with irregular dates
3. IRR(CashFlows) / XIRR(CashFlows, Dates) → Internal rate of return
4. PMT(rate, nper, pv) → Debt schedule
5. PV(rate, nper, pmt) / FV(rate, nper, pmt) → Present/future values

Step 7: Logical & Lookup Functions

1. IF(condition, value_if_true, value_if_false) → Conditional logic
2. IFS() → Multiple condition checks
3. AND() / OR() → Combine conditions
4. VLOOKUP() / HLOOKUP() → Fetch data from tables
5. INDEX() + MATCH() → Flexible dynamic lookup
6. XLOOKUP() → Modern, versatile lookup

Step 8: Valuation

1. **DCF**:
    - Free Cash Flow = Net Income + Depreciation - CapEx - ΔWC
    - Enterprise Value (EV) = NPV(WACC, FCFs)
    - Equity Value = EV - Debt + Cash
2. **Comps**:
    - Apply peer multiples (EV/EBITDA, P/E)
3. **Precedent Transactions**:
    - Apply historical deal multiples
4. **LBO (if needed)**:
    - Debt financing schedule → Equity IRR via XIRR()

Step 9: Scenario & Sensitivity

1. Use **Data Tables** → Test impact of 1–2 key variables
2. Use **Scenario Manager** → Compare Base / Optimistic / Pessimistic
3. Use **Goal Seek** → Solve for input to reach target (e.g., IRR = 15%)

Step 10: Error Checking & Auditing

1. Check Assets = Liabilities + Equity → flag errors
2. Check Cash Flow reconciliation → Opening + Net Cash = Closing
3. Use IFERROR() for clean formulas
4. Use Trace Precedents / Dependents → verify links

Step 11: Formatting & Documentation

1. Color code:
    - Inputs = Blue, Formulas = Black, Links = Green
2. Add comments or footnotes for assumptions
3. Freeze top rows for timeline visibility
4. Create summary dashboard for key outputs: Revenue, EBITDA, Net Income, EV, IRR

Step 12: Review & Stress Test

1. Check for negative cash balances or inconsistencies
2. Verify circular references (only if intentional)
3. Test extreme scenarios with sensitivity tables

KEY TOOL CONCEPTS IN FINANCIAL MODELLING

A. Core Financial Modelling Structure

1. Three-Statement Model

This is the heart of financial modelling, integrating:

- Income Statement (P&L): Projects revenue, expenses, and net income.
- Balance Sheet: Shows assets, liabilities, and equity at a point in time.
- Cash Flow Statement: Reconciles profit with actual cash movement.

All three are linked dynamically, meaning changes in assumptions automatically update all statements (e.g., depreciation affects net income, accumulated depreciation, and cash flow).

1. **Drivers and Assumptions Sheet**

This is a **central hub** where all key model inputs are stored, such as:

- Revenue growth rates, gross margins, and operating expenses.
- Working capital days (receivables, payables, inventory).
- CapEx, financing assumptions, and tax rates.

By keeping assumptions in one place, you make the model **transparent, flexible, and easy to update**.

2. **Scenario and Sensitivity Analysis**

Used to **test different outcomes** based on changing key variables:

- Scenario Analysis — Models different cases (e.g., Base, Bull, Bear).
- Sensitivity Analysis — Tests impact of one variable (e.g., WACC, terminal growth rate) on valuation.

This helps in assessing **risk and upside/downside potential**.

3. **Error-Checking & Audit Controls**

Implements **controls and flags** to ensure accuracy and reliability:

- Balance sheet checks (Assets = Liabilities + Equity).
- Cash flow reconciliation checks (Opening + Net Cash = Closing Cash).
- Error flags using conditional formatting.
- Manage circular references (e.g., interest expense based on average debt).

These controls **increase model integrity and credibility**.

4. **Timeline Setup**

   Defines the **time axis** for projections — monthly, quarterly, or annual — depending on the purpose.
   Proper setup allows for:

   - Consistent forecasting across all statements.
   - Smooth roll-forward of balances and calculations.

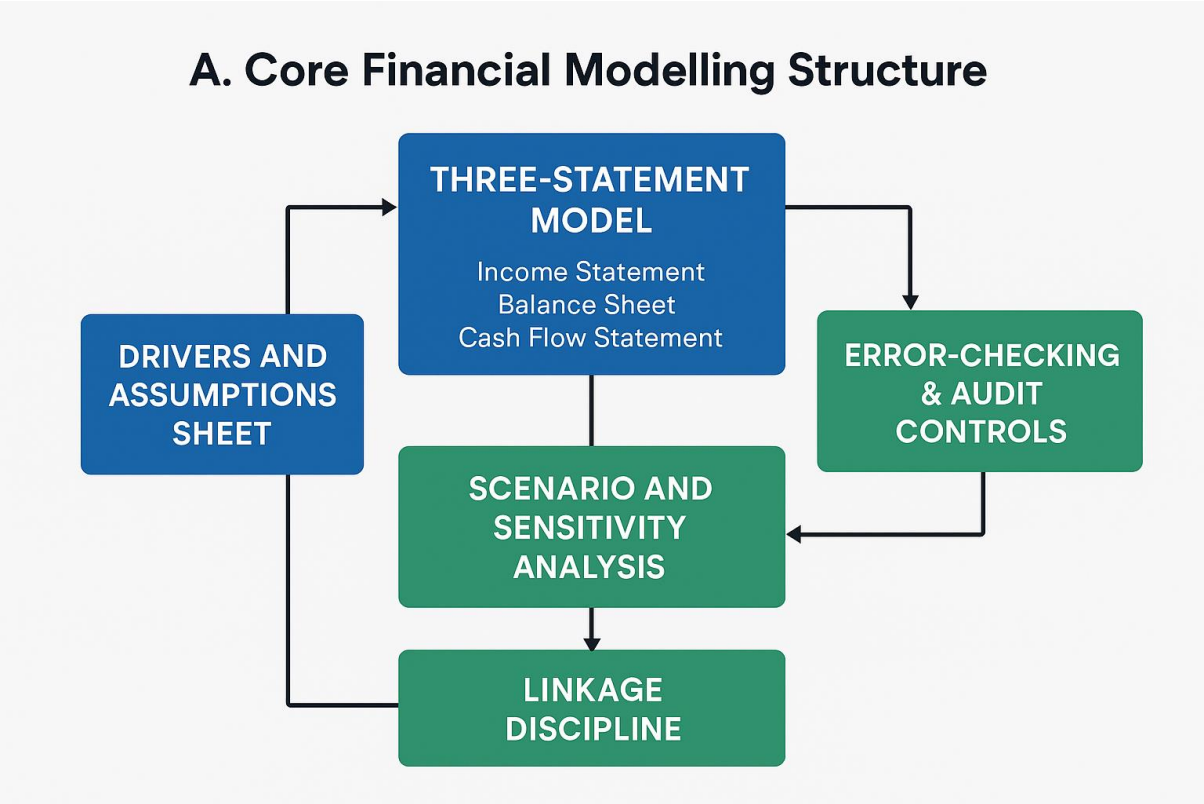
   A well-designed timeline ensures all projections align perfectly in time.

5. **Linkage Discipline**

   A key modelling best practice: **no hardcoding numbers inside formulas**.

   - Always link formulas to the assumptions sheet.
   - This allows for full traceability and easy scenario updates.
   - Hardcoding breaks transparency and creates hidden errors.

   Maintaining linkage discipline makes your model **auditable, flexible, and professional**.



B. Valuation Techniques

1. **Discounted Cash Flow (DCF)**

    - A fundamental (intrinsic) valuation method.
    - Values a business based on the present value of its future Free Cash Flows (FCF).
    - Cash flows are discounted using the Weighted Average Cost of Capital (WACC) to reflect time value and risk.
    - Produces an estimate of enterprise value (EV) and equity value after adjusting for debt and cash.

2. **Comparable Company Analysis (Comps)**

    A relative valuation approach comparing the target company to similar publicly traded peers.

    - Uses valuation multiples such as:
        - EV/EBITDA
        - EV/EBIT
        - P/E (Price/Earnings)
        - EV/Revenue
    - The company's implied value is derived by applying peer multiples to its own financial metrics.

    Purpose: To estimate how the market values similar businesses today.

3. **Precedent Transactions**

    - Also a relative valuation method, but based on historical M&A deals.
    - Examines what multiples were paid in comparable acquisitions (e.g., EV/EBITDA, EV/Sales).
    - Typically produces higher valuation multiples than trading comps due to control premiums paid by acquirers.

    Purpose: To estimate what a buyer might realistically pay in an acquisition scenario.

4. **Leveraged Buyout (LBO)**

    - A private equity-style valuation focusing on investor returns.
    - Assumes a business is bought primarily using debt (leverage) and later sold after several years.
    - Calculates Internal Rate of Return (IRR) and cash-on-cash multiple based on the equity investor's capital structure and exit assumptions.

    Purpose: To determine if the acquisition meets the investor's **target return** thresholds.

Together, these techniques provide a **comprehensive view of value** —

- **DCF** gives *intrinsic* value,
- **Comps** and **Precedents** give *market-based* value,
- **LBO** assesses *financial sponsor perspective*.

2. ESSENTIAL EXCEL FUNCTIONS FOR FINANCIAL MODELLING

A. Financial Functions

NPV()

Calculates the **Net Present Value** of future cash flows based on a constant discount rate. Assumes cash flows occur at regular intervals.

Example & Use:

=NPV(WACC, FCFs) → Discounts Free Cash Flows (FCFs) using the company's Weighted Average Cost of Capital (WACC).

XNPV()

Similar to NPV(), but allows **cash flows with specific dates** — making it more accurate for real-world timing differences.

Example & Use:

=XNPV(10%, cash_flows, dates) → Preferred in professional valuation models.

IRR()

Computes the **Internal Rate of Return** — the discount rate at which the NPV equals zero. Assumes regular time intervals.

Example & Use:

=IRR(cash_flows) → Measures project or investment return.

XIRR()

An enhanced version of IRR() that handles **irregularly spaced cash flows** using actual dates.

Example & Use:

=XIRR(cash_flows, dates) → Used in project finance, private equity, and DCF models.

PMT()

Calculates **loan or lease payments** per period based on interest rate, number of periods, and present value.

Example & Use:

=PMT(rate, nper, pv) → Used for debt schedules or amortization tables.

PV() / FV()

Determine the Present Value (PV) or Future Value (FV) of a series of payments or returns.

Example & Use:

=PV(rate, nper, pmt) → Finds today's value of future cash flows. =FV(rate, nper, pmt) → Projects future value of investments.

B. Logical and Lookup Functions

IF() / IFS()

Apply **conditional logic** to return different results depending on criteria. IFS() allows multiple conditions without nesting many IFs.

Example & Use

=IF(A1>0, "Profit", "Loss") → Returns "Profit" if A1 > 0, otherwise "Loss." Useful for flags, error checks, or scenario triggers.

AND() / OR()

Combine multiple conditions into a single logical test. Often used inside IF() statements.

Example & Use

=AND(A1>0, B1<100) → Returns TRUE only if both conditions are met. Helpful for complex assumptions or validation rules.

VLOOKUP() / HLOOKUP()

Look up data from a **vertical or horizontal** table. VLOOKUP() searches down the first column of a table and returns a value from a specified column.

Example & Use

=VLOOKUP(code, table, col_index, FALSE) → Retrieves matching data like product names or financial metrics.

INDEX() + MATCH()

More **flexible and powerful lookup** combination than VLOOKUP(). Works left-to-right or right-to-left and is less prone to break when columns move

Example & Use

=INDEX(values, MATCH(key, lookup_range, 0)) → Finds a value dynamically. Common in dynamic financial models

XLOOKUP() ***(modern Excel)***

The **new all-in-one lookup** function — replaces VLOOKUP, HLOOKUP, and INDEX-MATCH. Allows exact or approximate matches, two-way lookups, and error handling.

Example & Use

=XLOOKUP(lookup_value, lookup_array, return_array) → Clean, flexible, and preferred for modern models.

C. Data Analysis & Scenario Tools

Data Tables

Used for **sensitivity analysis** — shows how a result (e.g., valuation, IRR, EPS) changes when one or two key assumptions (like growth rate or WACC) vary. Helps visualize risk and impact on outcomes.

Goal Seek

A **back-solving tool** that finds the input value needed to reach a desired result. Example: determine what revenue growth is required for IRR = 15%.

Scenario Manager

Allows creation and comparison of **multiple sets of assumptions** (e.g., Base, Optimistic, Pessimistic) without manually changing input cells. Useful for management summaries.

Solver Add-in

Solves optimization problems by adjusting multiple variables under constraints. Example: find the optimal capital structure or minimize financing costs while maintaining target ratios.

PivotTables

Used to **summarize, analyze, and group large datasets** quickly — ideal for historical financial analysis, KPIs, or summarizing transaction data

D. Date & Time Functions

Function	Purpose / Description	Example & Use
TODAY()	Returns the **current date** (updates automatically each day).	=TODAY() → Useful for stamping report dates or calculating days remaining.
NOW()	Returns the **current date and time**.	=NOW() → Helpful for time-based updates or logging activity timestamps.
DATE(year, month, day)	Creates a date from individual year, month, and day values.	=DATE(2025,10,23) → Outputs **23-Oct-2025**. Common for timeline setup.
EDATE(start_date, months)	Returns a date a specified number of **months before or after** a given date.	=EDATE(A1, 12) → One year after date in A1. Useful for monthly forecasting.
EOMONTH(start_date, months)	Returns the **last day of the month** before or after a given date.	=EOMONTH(A1, 0) → Month-end date for A1. Ideal for monthly model periods.
YEARFRAC(start_date, end_date)	Calculates the **fraction of a year** between two dates.	=YEARFRAC(A1, B1) → Used in interest or depreciation calculations.
DATEDIF(start_date, end_date, unit)	Measures the **difference between two dates** in years ("Y"), months ("M"), or days ("D").	=DATEDIF(A1, B1, "M") → Months between two dates.
YEAR() / MONTH() / DAY()	Extracts the year, month, or day from a date.	=YEAR(A1) → Useful for grouping or timeline creation.
WEEKDAY()	Returns the day of the week as a number (1 = Sunday by default).	=WEEKDAY(A1) → For scheduling or business day logic.

E. Error Handling & Auditing

IFERROR()

Prevents errors like #N/A or #DIV/0! by allowing you to display a custom value instead of the error. Example: =IFERROR(A1/B1, "Error").

ISNUMBER() / ISTEXT()

Checks the data type of a cell. ISNUMBER() returns TRUE if the cell contains a number; ISTEXT() returns TRUE if it contains text. Useful for validation before calculations.

TRACE DEPENDENTS / PRECEDENTS

Auditing tools that visually show which cells affect (precedents) or are affected by (dependents) a selected cell. Helps track formulas and detect errors.

F. Dynamic & Array Functions (Modern Excel)

Function	Purpose
FILTER()	Returns a dynamic array of data that meets specified criteria. Example: =FILTER(A2:A10, B2:B10="Yes").
SORT()	Sorts a range or array dynamically without manually rearranging data. Example: =SORT(A2:A10,1,TRUE) sorts ascending.
UNIQUE()	Extracts distinct values from a range or array. Example: =UNIQUE(A2:A10).
SEQUENCE()	Generates a series of numbers (or dates) dynamically. Example: =SEQUENCE(5,1,1,1) creates 1,2,3,4,5.
LET() / LAMBDA()	LET() assigns names to calculation parts to simplify formulas. LAMBDA() lets you create reusable custom functions. Example: =LET(x, A1*2, x+5).

3. Pro Tips for Professional Modelling

- **Avoid hardcoding:** Place all assumptions in one sheet.
- **Use consistent formatting:** Inputs (blue), formulas (black), links (green).
- **Balance the model:** Always check *Assets* = *Liabilities* + *Equity*.
- **Name ranges:** For readability (e.g., =Revenue_Growth).
- **Document assumptions:** Add comments or an "Assumptions" section.
- **Stress test your model:** Check for negative cash or circular references.