**Task: Energy Flow Optimization**

Design and implement an optimization model for an energy system consisting of a photovoltaic (PV) system, an electrical battery, and a connection to the external electrical grid.The objective is to meet predicted electrical energy demands while minimizing costs.

The connections within the system are presented in the scheme. Each arrow represents an energy flow. All energy flow and system characteristics are in kW or kWh units.

A diagram of electrical grid

Description automatically generated

**System Description**

* **Photovoltaic (PV) System**: Produces electricity in kWh (no losses).
* **Battery**:
  + Capacity: 160 kWh
  + Max charge/discharge rate: 100 kW
  + Charging efficiency: 92% (discharging is lossless)
  + Levelized cost of storage (LCOS): Consider that each kWh discharged from the battery adds a fixed cost.
  + Initial charge: 0 kWh
* **Electrical Grid**:
  + Max sell power: 700 kW
  + Max buy power: 700 kW
* **Consumer**: Must be supplied each hour with a specified energy demand.

You will receive the file with hourly pv production, electrical consumption and energy prices.

1. Implement a linear optimization model that minimizes:

Total Cost = ∑ (buy\_price × energy bought) - ∑ (sell\_price × energy sold) + ∑ (lcos × energy discharged from battery)

**Optional Challenges**

**B)** Disallow buying and selling from the grid simultaneously.  
**C)** Only allow buy/sell in **100 kWh packages**. Allow battery capacity to be increased (at a cost).

Feel free to share your thoughts:

* What other real-world constraints would you add?
* How would you adapt your model for longer horizons or minute-level timesteps?

**Deliverables**

* Python Code (Pyomo suggested)
* Plots of energy flows per component per hour
* Final cost and summary table
* Description of assumptions and model
* Optional: Include answers to B/C as a markdown or PDF

**Submission**

Upload your solution to a public GitHub repository or share as a ZIP file. Include a README.md with:

* Setup instructions
* Model explanation
* Notes on optional parts