Solar Storage Simulation Plan

Project Goal

Create an accurate household energy storage simulation to replace the current lookup table approach with real-time calculations using 15-minute interval consumption data.

Current System Analysis

How the Lookup Tables Work (Simple Summary)

The current system uses a **2D lookup table** (`matrix_ev_quote`) to estimate the **self-consumption rate** (Eigenverbrauchsquote) of a PV system:

Simple Explanation:

- **Input**: How big is your battery vs. your yearly energy use? How big is your PV system vs. your yearly energy use?
- **Output**: What percentage of your solar energy can you use yourself (instead of feeding into the grid)?

Example:

- Family uses 5,000 kWh/year
- Has 10 kWh battery → ratio = 10/5 = 2.0
- Has 8 kWp PV system → ratio = 8/5 = 1.6
- Lookup table says: ~65% self-consumption rate
- Multiply by consumption pattern factor (day/night/optimized): 65% × 1.1 = 71.5%

Detailed Technical Explanation

Matrix Structure

```
matrix_ev_quote[row][column] = base self-consumption percentage

Row Index = 165 - (battery_kWh / annual_consumption_MWh)
Column Index = (PV_size_or_yield / annual_consumption_MWh)

Where:
- Battery ratio: kWh per MWh/year consumption
- PV ratio: kWp per MWh/year (household) OR kWh/year per MWh/year (heat
pump)
```

Consumption Pattern Adjustments

The base lookup value is multiplied by consumption characteristic factors:

Pattern	Household Factor	Heat Pump Factor	E-Car Factor
Day distributed	1.1	0.72858	0.61105
Evening/Morning	1.0	0.4839	0.5555
Yield optimized	1.18	0.77424	0.65549

Current Limitations

1. **Matrix bounds**: If ratios exceed table dimensions, system fails or uses 3% fallback
2. **Generic assumptions**: One-size-fits-all approach doesn't reflect real consumption patterns
3. **No temporal matching**: Doesn't consider when energy is produced vs. consumed
4. **Static factors**: Adjustment factors are fixed, not dynamic

What We Need to Calculate/Simulate

1. Energy Production Simulation

- **Solar irradiation data**: Hourly/15-minute intervals for full year
- **PV system parameters**: Size (kWp), tilt, azimuth, efficiency factors
- **Weather factors**: Temperature effects, shading, inverter efficiency
- **Output**: kWh produced every 15 minutes for entire year

2. Energy Consumption Simulation

- **Household base load**: 15-minute consumption patterns from standardlastprofil-haushaltskunden-2026.xlsx
- **Heat pump consumption**: From 2025-05-27_Wärmepumpe_Lastgänge.xlsx
- **E-car consumption**: From Standardlastprofile_Elektrofahrzeuge_Anhang_E.xlsx
- **Scaling factors**: Based on user's annual consumption input
- **Output**: kWh consumed every 15 minutes for entire year

3. Storage Simulation Logic

```
For each 15-minute interval:
    production = solar_kwh_produced
    consumption = total_household_kwh_needed

    if production > consumption:
        excess = production - consumption
        if storage_current < storage_capacity:
            # Fill storage first
            storage_add = min(excess, storage_capacity - storage_current)
            storage_current += storage_add
            excess -= storage_add

        # Remaining excess goes to grid (get paid)
        grid_feed_in += excess
```

```
    else:  # consumption > production
        deficit = consumption - production
        if storage_current > 0:
            # Use storage
            storage_use = min(deficit, storage_current)
            storage_current -= storage_use
            deficit -= storage_use

        # Remaining deficit comes from grid (pay for it)
        grid_consumption += deficit
```

4. Economic Calculation

- **Self-consumption value**: kWh × electricity_price
- **Feed-in revenue**: kWh × feed_in_tariff
- **Grid purchase cost**: kWh × electricity_price
- **Net savings**: Self-consumption value + Feed-in revenue - Grid purchase cost

Input Variables (From Frontend Website of the Adrex Konfigurator (only examples not all parameters are included yet, f.e. only a few module types like winiaco are added as examples))

Location & System Configuration

- **Postleitzahl**: 5-digit postal code (integer)
- **PV Module count**: Integer (e.g., 40 modules)
- **PV Module type**: Dropdown selection affecting Wp per module
- **Battery count**: Integer (e.g., 0-10+ batteries)
- **Battery type**: Dropdown affecting kWh per battery
- **Tilt angle**: Integer degrees (0-90°)
- **Azimuth**: Integer degrees (-180° to +180°)

Consumption Profile

- **Household size**: 1-5+ persons affecting base consumption
- **Annual consumption**: kWh/year (integer)
- **Consumption pattern**: Day/Evening/Optimized (affects timing)
- **Heat pump**: Yes/No + annual consumption if yes
- **E-car**: Yes/No + annual consumption if yes

Advanced Parameters

- **Inverter efficiency**: Percentage (default ~96%)
- **System losses**: Percentage (default ~20% total)
- **Degradation**: Annual percentage (default ~0.5%)
- **Shading factors**: Percentage losses (or more accurate shading factors, if google API is used (to use Google API the Postleitzahl isnt enough, we would need exact coordinates of the house))

Simulation Implementation Plan

Phase 1: Data Preparation

1. **Load consumption profiles** from Excel files
2. **Create scaling functions** to match user's annual consumption
3. **Validate data completeness** (8760 hours × 4 = 35,040 intervals)

Phase 2: Production Engine

1. **Enhance current solar calculator** with all frontend parameters that can be inputed, modules etc. (to find in Admin Page of the Adrex Configurator)
2. **Generate 15-minute production data** for full year
3. **Apply all efficiency factors** dynamically
4. **Include temperature and shading effects**

Phase 3: Storage Simulation Engine

1. **Implement 15-minute storage logic**
2. **Track storage state** throughout the year
3. **Calculate grid interactions** (feed-in and consumption)
4. **Generate detailed energy flows**

Phase 4: Results & Validation

1. **Calculate annual totals** and compare with lookup table results
2. **Generate detailed breakdowns** by month/season
3. **Provide economic analysis (optional)** with different tariff scenarios

Phase 5: Optimization & Integration into app

1. **Performance optimization** for large datasets
2. **API interface** for TypeScript integration; include calcualtor into Frontend

Expected Improvements Over Lookup Tables

1. **Accuracy**: Real consumption patterns vs. generic assumptions
2. **Temporal matching**: Considers when energy is produced and consumed
3. **Dynamic factors**: Adjusts for actual weather, temperature, usage patterns
4. **Detailed insights**: Shows monthly/seasonal variations
5. **Scenario analysis**: Easy to test different configurations
6. **No bounds limits**: Works for any system size combination
7. **Transparency (if wanted)**: Users can see exactly how calculations work

Data Sources Required

1. **Solar irradiation**: Already available in optimized 500MB grid (data will be downloaded from PVGIS API, or different API? API has still to be decided on)
2. **Consumption profiles**: Available in Excel files in modeling folder
3. **Weather data**: Temperature effects (can be estimated or enhanced later; if only long periods are calculated and not single days, weather data as well as live solar irridation data, isnt nessesary, f.e. for

a year the solar irridation and weather data from the last years will suffice)
4. **Tariff data (if economic calculations are also included in this calculator)**: Electricity prices and feed-in rates (user input)

Success Metrics

1. **Accuracy validation**: Results within ±5% of other sources/calculators for standard cases
2. **Performance**: Full year simulation in <10 seconds
3. **Coverage**: Handles edge cases that break lookup table
4. **Usability**: Clear, detailed results that explain the calculations
5. **Flexibility**: Easy to adjust parameters and see immediate impact

---

This simulation will provide the foundation for accurate, transparent, and flexible solar storage analysis that can replace the current lookup table approach with real-world consumption patterns.