

Programmbeschreibung

Eingabe:

Nutzer*innen können über den Link zu einem Streamlit Programm geleitet werden, die über einen Cloud-Service gehostet wird. Streamlit ist ein Python Paket, dass Python Skripte über ein Browser-Fenster zugänglich macht und Eingabe- als auch Visualisierungsmöglichkeiten bietet.

Die Nutzer*innen werden in der Webapp dazu aufgefordert sämtliche Eingaben zu machen, die zu der Berechnung der individuellen CO2-Kompensation genutzt werden können.

Die wichtigsten **Parameter** sind:

- Der jährliche Strom- (*annual_elec_demand*) und Wärmebedarf (*annual_heat_demand*)
- Der Standardlastprofil-Typ, i.e., ob die Nutzer*in in einem Einfamilien- (EFH) oder Mehrfamilienhaus (MFH) wohnt. In der Zukunft kann das Tool auch dazu eingesetzt werden CO2-Kompensation für ein Gewerbe zu berechnen, aber das ist derzeit nicht vorgesehen.
- Die Koordination, i.e. Breitengrad (*lat* von Latitude) und Längengrad (*lon* von Longitude)
- Das Bundesland (*province*), da das Tool derzeit für Deutsche Nutzer*innen ausgelegt ist)
- Der „Strom-Mix“ (*elec_mix*), i.e., der prozentuale Anteil von Ökostrom im individuellen Tarif, was zur Berechnung der CO2-Belastung pro kWh Strom eingesetzt wird.
- Die eingesetzte Heiztechnologie (*heat_tech*), was zur Berechnung der CO2-Belastung pro kWh Wärme (bzw. Gas) eingesetzt wird.
- Das Beheizungssystem (*heat_system*), also ob eine Fußbodenheizung (FBH) oder ein Heizkörpersystem (HKS) genutzt wird, was für die Berechnung der Vorlauftemperatur der Heizung eingesetzt wird und damit ausschlaggebend ist für die Heizungsleistung der Heiztechnologie.
- Das Temperaturziel für die Auswahl des dazugehörigen CO2-Preises (*co2_price_sim*)
- Die Parameter zur Bestimmung der Potenziale für erneuerbare Energien:
 - o Eine mögliche Photovoltaik Fläche zur Stromerzeugung (*p_v_area*)
 - o Einer von drei möglichen Wärmepumpen Konfigurationen (*heat_pump*), die jeweils aus der Wärmequelle und dem Wärmeträgermittel bestimmt ist:
 - Luft-Wasser Wärmepumpe (*HP_Air*)
 - Sole-Wasser Wärmepumpe (*HP_Ground*)
 - Wasser-Wasser Wärmepumpe (*HP_Water*)

Jeder dieser Wärmepumpen weist individuelle Effizienz- bzw. Verlustparameter auf.

 - o Eine mögliche Kollektorfläche zur Nutzung der Solarthermie (*st_area*) und der ausgewählte Kollektortyp (*st_collector*).
 - o Die Nabenhöhe für eine kleine Windkraftanlage (*hub_height*), obwohl derzeit – aufgrund der schlechten Datenlage – keine Windenergienutzung vorgesehen ist.

Programmablauf

(*main_submit*)

1. Der CO₂-Preis für die Berechnung. Dieser wird anhand des gewählten Temperaturziels initialisiert. Dies geschieht mittels der Funktion *co2_price* in der Datei *calc.py*.
 - Es stehen vier Szenarien zur Auswahl: *BAU* (business-as-usual), *Paris1.5* und *Paris2* (analog für die jeweiligen Temperaturziele, die in dem Paris-Vertrag festgelegt sind) und *Neutral45* (was einem nationalen CO₂-Preis entspricht, der mit einem Entwicklungspfad vereinbar ist auf dem Deutschland bis 2045 Klimaneutral sein würde).
2. Die Wetterzeitreihen (*weather_hourly*). Wird mittels der Funktion *tmy_data* Wetterdaten für ein typisches meteorologisches Jahr (Abk.: *tmy*) abgerufen für den eingegebenen Standort
3. Die Strom- und Wärmebedarfszeitreihen für die jeweiligen eingegebenen jährlichen Nachfragewerte. Diese werden mit der *elec_slp* bzw. *heat_slp* Funktion berechnet, welche für das gewählte Simulationsjahr (*year*) der Nutzer*in auf das *demandlib* Paket zugreift.
 - Der Index der erzeugten Lastgänge werden unter *time_index_year* gespeichert, sodass andere Zeitreihen (insb. Einspeisezeitreihen) auf diesem zugreifen können. Das soll dazu dienen, dass später die ausgegebenen Zeitreihen der Ergebnisse alle den gleichen Zeitraum abbilden.
4. Die Einspeisezeitreihen der erneuerbaren Energien (**Strom**) – *feedin_fn*
 - Photovoltaik (PV) – *pv_elec*
 - Es wird eine Konstellation aus einem „Silevo Triex“ Modul mit einer 300 W Ausgangsleistung und einem „ABB Micro“ Inverter mit Niederspannungsausgang (240 V) genutzt. Dieses Paar ist repräsentiert ein fortgeschrittenes Setup zur Stromerzeugung.
 - Die Ausrichtung (*azimuth*) ist exakt südlich (180°) mit einer „idealen“ 37,5° Neigung (*tilt*).
 - Es wird ein *ac* Modus gewählt, der die Verluste des Inverters miteinbezieht. Die Flächenskalierung (*area scaling*) dient dazu die Erzeugung auf den Quadratmeter runterzurechnen, sodass die Gesamterzeugung durch Multiplikation mit der eingegebenen PV-Fläche berechnet wird. Da die Ausgabe in Watt erfolgt, muss auf kW mittels einer 1000-Teilung umgerechnet werden.
 - **Empfehlung:** Für Nutzer*innen die bereits eine PV-Anlage bei sich zuhause stehen haben, sollte man eine detailliertere Eingabe erlauben für Anlagen-spezifische Werte.
 - Wind ist derzeit nicht implementiert (daher wird nur ein 0-Vektor initialisiert) sollte aber in Zukunft (mit einer besseren Datenbasis) möglich sein.
 - Wichtige „Infrastruktur“ wie z.B. das Einlesen einer Leistungskurve und Leistungskoeffizienten Kennlinie sind bereits implementiert (siehe die Funktion *wind_elec*)
 - **Empfehlung:** Verlässliche (und einfach abrufbare) Daten finden und einbauen!

5. Die Einspeisezeitreihen der erneuerbaren Energien (**Wärme**) – *thermal_fn*

- Solarthermie (*st_feedin*)
 - Die Berechnung der Solarthermie Wärmeerzeugung erfolgt in der Funktion *soltherm_heat*. Durch die Nutzer*in wird festgelegt welcher Kollektortyp ausgewählt wird für die Erzeugung.
 - Es stehen zwei „repräsentative“ Kollektortypen (für die nicht-konzentrierte Solarthermie) zur Auswahl: ein Vakuumröhrenkollektor und ein Flachkollektor. Die technischen Parameter (wie z.B. optischer Wirkungsgrad) sind in dem Excel Datenblatt *soltherm_data* zu finden.
 - Auch hier erfolgt eine Berechnung in Watt, die zu Kilowatt umgerechnet wird.
- Wärmepumpe
 - **Wichtig:** In diesem Tool wird die Annahme getroffen, dass Haushalte ihre CO₂-intensiven existierenden Heiztechnologien (wie den Gaskessel oder die (noch) Gas-belastete Fernwärme) *komplett* mit einer Wärmepumpe ersetzen. Daher wird nicht die Wärmeerzeugung der Wärmepumpe berechnet, sondern der zusätzliche Strombedarf, der entsteht, wenn die Wärmepumpe die Wärmenachfrage deckt, die *nach* der Einbindung der Solarthermie noch besteht. Hierzu wird die Leistungszahl (*coefficient of performance*, Abk.: *COP*) genutzt und das mathematische Verhältnis:
$$P_{el} = \frac{Q_{Heiz}}{COP}$$
 - Über die Wahl des Heizungssystems (Fußbodenheizung *FBH* oder Heizkörper *HKS*) wird die Vorlauftemperatur bestimmt (*T_out*), die maßgeblich ist für die Leistung der Wärmepumpe. Sowohl die Temperatur der Wärmequelle (*T_in*) als auch der Nutzungsgrad der Anlage (*quality_grade*) ist abhängig von der gewählten Konfiguration (i.e. Luft-Wasser, Wasser-Wasser oder Sole-Wasser).
 - Letztendlich wird über das Verhältnis *heat_demand_renewable* (i.e. inkl. Solarthermie) geteilt durch *cop*, der elektrische Verbrauch der Wärmepumpe berechnet (*heat_pump_el*)
 - **Empfehlung:** Aufgrund eines Fehlers bei der Funktion *calc_cops* kann derzeit nur eine **konstante** Leistungszahl angenommen werden, die einer

6. Die CO₂-Emissionen

- Die CO₂-Emissionen werden allgemein mittels der *co2_kWh* berechnet (in dem *calc.py* Skript). Diese liest den Emissionsfaktor der Technologie aus dem Datenblatt ein und multipliziert den stündlichen Verbrauch bzw. Erzeugung mit dem Emissionsfaktor
- Es werden die „alten“ Emissionen des Stromverbrauchs (*co2_old_elec*) mit den „neuen“ (*co2_new_elec*) verglichen, um die Kompensation zu bestimmen.

- Die „neuen“ CO₂-Emissionen setzen sich aus Emissionen zusammen, die durch die Stromerzeugung entstehen (*ren_tech_co2*, bezogen auf *Life-Cycle-Assessment* Werte) und die, die durch den Netzbezug entstehen, welches die Residuallast decken muss.
 - Die Residuallast ist die Differenz aus dem Strombedarf (inkl. Wärmepumpenlast) und der Einspeisung aus den Erneuerbaren.
- Aufgrund der getroffenen Annahmen zur Wärmepumpe, wird der „neue“ Gasverbrauch mit einem 0-Vektor initialisiert. Somit sind die einzigen CO₂-Emissionen, die durch den Gaseinsatz entstehen, den „alten“ Erzeugungstechnologien zuzuschreiben (*co2_old_gas*).
- Die **Kompensation** wird berechnet, indem alle neuen und alten CO₂-Werte summiert werden und die neuen von den alten abgezogen werden:

$$comp = old - new$$

Somit bedeutet ein negativer Wert, dass die „neue“ Erzeugungskonstellation **mehr** CO₂-Emissionen verursacht als vorher. Ein positiver Wert ist genau die Menge an CO₂-Emissionen, welche eingespart (bzw. kompensiert) worden sind (in dem gegebenen Zeitraum).

7. Die Ersparnisse bzw. die Mehr-Kosten

- Analog zur Berechnung der CO₂-Emissionen wird für jede Stunde, wurde eine *price_kWh* Funktion eingesetzt, die für die alten und neuen Strom- und Gasbedarfszeitreihen den Preis pro kWh berechnet, *ohne* CO₂-Preis. Die Ersparnisse bzw. Mehrkosten werden in *price_diff* berechnet (genau wie die Kompensation auch)
- *total_diff* berechnet dann die Ersparnisse bzw. Mehrkosten **mit** einem CO₂-Preis.
- **Wichtig:** Viele erwarten, dass besonders inkl. einem CO₂-Preis, die Berechnung *immer* **Gesamtersparnisse** ausgibt. Das muss aber nicht zwingend der Fall sein. Z.B. wenn der Wärmebedarf eines Haushalts immens ist, und dieser durch eine Wärmepumpe gedeckt werden muss, kann der Stromtarif (und dessen CO₂-Belastung) eine große Auswirkung haben und Preise bzw. CO₂-Emissionen in die Höhe schießen!
- **Empfehlung:** Derzeit wird angenommen dass jeder Haushalt bei seinem „alten“ Tarif bleibt. Stattdessen, sollte entweder die Option ermöglicht werden auf einen Ökostrom Tarif zu wechseln (falls dieser noch nicht von der Nutzer*in gewählt wurde), oder einen „custom“ Stromtarif festzulegen!