Technical University of Denmark

Solpanel projekt
Gruppenavn: Solpanel10

Course number: 01002

Authors

Student - s223750 - Amalie Rand
Student - s234811 - Asger Thousig Sehested
Student - s234825 - Benedicte Lumby
Student - s234800 - Elias Lyngsberg Rasmussen
Student - s234873 - Philip Kierkegaard
Student - s234846 - Rasmus Arnmark

1. maj 2024

Indhold

Introduktion **2**

Indledende øvelser **2**

Solpositionsmodel **3**

Solpositionsmodellering ved Pvlib **8**

Effekt og energiberegning **11**

Optimal vinkel **12**

Udvidelser **15**

Krum flade . . 15

Klog energiproduktion . . 20

Konklusion **23**

Introduktion

I 2022 udgjorde solenergi 6,2% af Danmarks samlede energiforbrug[1]. Dette tal forventes at stige i takt med høje energipriser og fokus på grøn omstilling. Danmarks nordlige breddegrader begrænser imidlertid solenergiens effektivitet i dele af året. Derfor er vedvarende forskning og udvikling afgørende for at øge solenergiens potentiale som en betydelig del af Danmarks energiforsyning. Dette projekt takler nogle af de matematiske problemstillinger og overvejelser, man skal gøre sig ved opsætning af solpaneler.

Indledende øvelser

Hver eneste time leverer solen mere energi til jorden, end alle lande samlet bruger på et helt år[2]. Der er altså kæmpe potentiale i solenergi som en bæredygtig og vedvarende energikilde. Solenergi udnytter solens elektromagnetiske stråling. Denne energiform er særligt attraktiv, da den tilbyder et bæredygtigt alternativ til fossile brændstoffer, og fordi den er i stand til at generere elektricitet uden emission af drivhusgasser. Teknologisk udvikling inden for solenergiområdet har væsentligt forbedret økonomisk levedygtighed og systemeffektivitet, hvilket gør solenergi til en stadig mere konkurrencedygtig energikilde på det globale marked.

Solceller, også kendt som fotovoltaiske celler (PV), er den primære teknologi anvendt til at konvertere sollys til elektricitet. Solceller virker ved at udnytte den fotovoltaiske effekt, hvor lysenergi (fotoner) fra solen absorberes af halvledermaterialer såsom silicium. Når disse fotoner absorberes, frigøres elektroner i materialet og skaber en strøm af elektriske ladninger, der fanges og ledes som elektricitet.

Solcellernes effektivitet afhænger af flere faktorer, herunder materialekvalitet, cellekonfiguration, og ikke mindst solcellernes evne til at absorbere lys. Monokrystallinske solceller, der består af en enkelt krystalstruktur, er kendt for deres høje effektivitet og evne til at fungere optimalt under svagt lysforhold, mens polykrystallinske solceller består af flere mindre siliciumkrystaller og generelt er billigere, men med en lavere effektivitetsrate.

I Danmark er den optimale vinkel for opstilling af solceller 38° stik syd, hvis man vil maksimere den årlige energiproduktion [3]. Vi tager i det følgende udgangspunkt i firmaet Trina Solar's solpanel Vertex S. Det måler 1762 x 1134 mm, og har under STC en Peak Power (Wp) på 425 W. STC er Standard Test Conditions og defineres her som irradians på 1100 $\frac{W}{m^2}$, temperatur på 25° og Air Mass AM = 1,5. Peak Power pr. kvadratmeter regnes som følgende: $\frac{425W}{1{,}762\text{m}\cdot 1{,}134\text{m}} = 212,7\frac{W}{m^2}$.

Under ideelle forhold står solens stråler vinkelret ned på panelet og solens irradians er konstant 1100 $\frac{W}{m^2}$ igennem en time. Når denne optimale irradians på $1100\frac{W}{m^2}$ opretholdes i en time, kan den samlede energi indfanget af et solpanel med et areal på $1,762\,\text{m} \cdot 1,134\,\text{m}$ beregnes. Ved at gange irradiansen over en time med panelets areal, opnår vi en enrgiproduktion på $1100\frac{W}{m^2} \cdot 1\,\text{h} \cdot 1,762\,\text{m} \cdot 1,134\,\text{m} = 2217,9\,\text{Wh}$, eller omkring $2,2\,\text{kWh}$. Over en hel time vil dette solpanel altså generere $2,2\,\text{kWh}$ af energi. Denne energimængde på $2,2\,\text{kWh}$ kan også udtrykkes i joule, da 1 Wh er lig med 3600 J. Således vil $2,2\,\text{kWh}$ være det samme som $2,2 \cdot 3.600.000\,\text{J}$, hvilket giver $7984\,\text{kJ}$.

Hvis vi ønsker at udtrykke denne energi pr. kvadratmeter for at sammenligne med irradiansen, dividerer

vi den samlede energi genereret på en time med solpanelets areal. Dette giver os $\frac{2217{,}9\,\mathrm{Wh}}{1{,}762\,\mathrm{m}\cdot 1{,}134\,\mathrm{m}} = 1100\frac{\mathrm{Wh}}{m^2}$. Dette viser, at under ideelle forhold, hvor panelet er orienteret perfekt i forhold til solen, er energiproduktionen per kvadratmeter lige så høj som den irradians, der modtages fra solen. Dette tager selvfølgelig ikke højde for at solpanelet ikke er 100% effektivt.

Herefter finder vi et udtryk for fluxen Φ, givet ved A_0, u_p, V, L og B. Først omskrives solens vektorfelt V til $V = u_s \cdot S_0$. Herefter kan vi integrere ved brug af definition 7.5.1 fra noterne.

$$\Phi = \int_0^L \int_0^B \langle u_s, u_p \rangle \cdot S_0 \cdot A_0 \, dB \, dL$$

Da hverken S_0, A_0 eller $\langle u_s, u_p \rangle$ afhænger af L eller B kan vi sætte disse udenfor integraltegnet:

$$\Phi = \int_0^L \int_0^B \langle u_s, u_p \rangle \cdot S_0 \cdot A_0 \, dB \, dL = \langle u_s, u_p \rangle \cdot S_0 \cdot A_0 \int_0^L \int_0^B dB \, dL$$

Vi får derfor at fluxen Φ kan udtrykkes som:

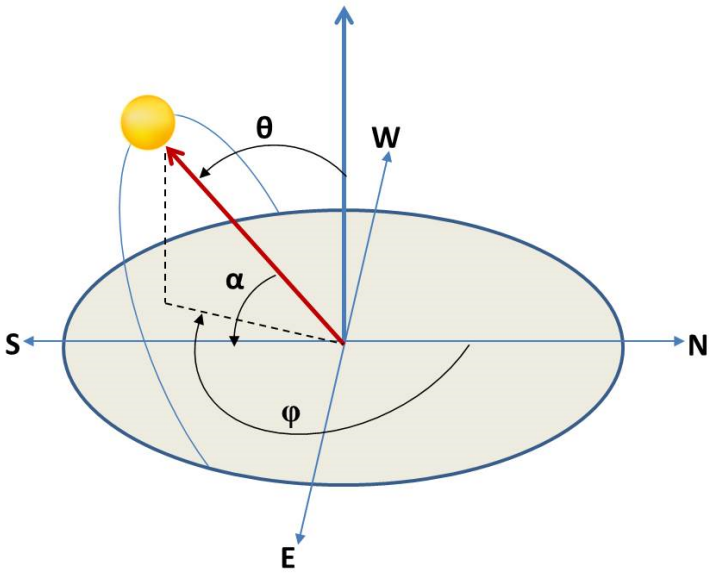
$$\Phi = \langle u_s, u_p \rangle \cdot S_0 \cdot A_0 \cdot L \cdot B, \quad 0 < \langle u_s, u_p \rangle < 1$$

Her er SI-enhederne for arealet LB, m^2, for V er det W/m^2. A_0 er en koefficient og har derfor ingen enhed. Fluxen Φ måles i W og energi måles i Joule.

Solpositionsmodel

Til at modellere solens position benytter vi det horisontale koordinatsystem, hvor z-aksen peger mod zenitpunktet i retning af normalvektoren til jordens tangentplan ved panelets position, x-aksen er rettet mod nord, og y-aksen mod øst. Solpanelet placeres i koordinatsystemets origo. Dette system afspejler observatørens lokale horisont og er orienteret efter dennes perspektiv.

Inden for dette system kan solens bevægelse og dens interaktion med solpanelet beskrives præcist. Vi anvender sfæriske koordinater, (r, θ, ϕ), hvor r er radius, $\theta \in [0, \pi]$ er zenit-vinklen fra z-aksen mod x-aksen og $\phi \in [0, 2\pi]$ er azimut-vinklen, der måles fra x-aksen mod y-aksen, til at repræsentere solens position. Det giver os mulighed for at beregne solens højdevinkel og dens afstand til solpanelet. Solens kartesiske koordinater transformeres til sfæriske for at forenkle beregningen af vinklen mellem solens normalvektor og solpanelets normalvektor. Denne vinkel er afgørende, da den bestemmer hvor effektivt solpanelet kan opsamle sollys – jo mindre vinklen er, desto mere direkte sollys modtager panelet.

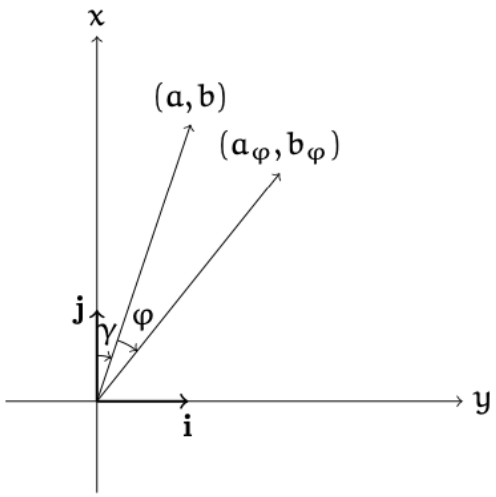


Figur 1: Det horisontale koordinatsystem

α er komplementærvinklen til zenit-vinklen, og kan findes ved følgende pythonkode:

```
def solar_elevation_angel(theta):
    return 90 - theta
```

Med udgangspunkt i de sfæriske koordinater og under antagelse af, at solens afstand til jorden er $r_s = 147.000.000.000$ m, vil vi nu udlede overgangen til de kartesiske koordinater. Først vil vi udlede rotationsmatricen i (y,x)-planet:



Figur 2: Rotation i (y,x)-planet

Rotationen, vi vil beskrive, er $\begin{bmatrix} a_\varphi \\ b_\varphi \end{bmatrix} = R_\varphi \cdot \begin{bmatrix} a \\ b \end{bmatrix}$, hvor R_φ er rotationsmatricen og rotationen måles fra x-aksen. Da matricer er lineære afbildninger, kan det skrives som $a \cdot R_\varphi \cdot j + b \cdot R_\varphi \cdot i$.

Om standard rotationsmatricen for (x,y)-planet gælder, at

$$R_\varphi i = \begin{bmatrix} cos(\varphi) \\ sin(\varphi) \end{bmatrix} \tag{1}$$

men da vi nu udleder rotationsmatricen for (y,x)-planet byttes første- og andenkoordinaten, da x nu er den vertikale akse, svarende til sinus og y er den horisontale akse svarende til cosinus, hvorved vi får:

$$R_\varphi i = \begin{bmatrix} sin(\varphi) \\ cos(\varphi) \end{bmatrix} \tag{2}$$

Det samme gøres for

$$R_\varphi j = \begin{bmatrix} cos(\frac{\pi}{2} + \varphi) \\ sin(\frac{\pi}{2} + \varphi) \end{bmatrix} \tag{3}$$

som nu bliver til

$$R_\varphi j = \begin{bmatrix} sin(\frac{\pi}{2} + \varphi) \\ cos(\frac{\pi}{2} + \varphi) \end{bmatrix} \tag{4}$$

vi kan nu bruge følgende trigonometriske identiteter på $R_\varphi j$:

$$sin(\tfrac{\pi}{2} + \varphi) = cos(\varphi)$$
$$cos(\tfrac{\pi}{2} + \varphi) = -sin(\varphi)$$

og får dermed

$$R_\varphi j = \begin{bmatrix} cos(\varphi) \\ -sin(\varphi) \end{bmatrix} \tag{5}$$

Da basisvektoren **i** beskriver y-aksen, og **j** beskriver x-aksen, og vinklen måles ud fra x-aksen vil første søjle i rotationsmatricen nu bestå af $R_\varphi j$ og anden søjle er $R_\varphi i$, grundet drejningen i urets retning. Rotationsmatricen for (y,x)-planet bliver altså

$$R_\varphi = \begin{bmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{6}$$

Beviset for disse trigonometriske identiteter findes i cosinus- og sinus-regneregler. En cosinusregneregel siger følgende:

$$cos(A - B) = cos(A) \cdot cos(B) + sin(A) \cdot sin(B) \Rightarrow \tag{7}$$

$$cos(\frac{\pi}{2} - \varphi) = cos(\frac{\pi}{2}) \cdot cos(\varphi) + sin(\frac{\pi}{2}) \cdot sin(\varphi) \tag{8}$$

$cos(\frac{\pi}{2})$ giver som bekendt 0, mens $sin(\frac{\pi}{2})$ giver 1. Derfor bliver udtrykket følgende:

$$cos(\frac{\pi}{2} - \varphi) = sin(\varphi) \tag{9}$$

Lignende regler gælder for $cos(A + B)$, $sin(A + B)$ og $sin(A - B)$. Vi vælger derfor ikke at bevise de andre trigonometriske identiter, da de følger direkte heraf.

Vi vil nu bruge dette til at udlede de kartesiske koordinater fra de sfæriske koordinater:

Vi fandt rotationsmatricen for (y, x)-planet til

$$R_\varphi = \begin{bmatrix} cos(\varphi) & sin(\varphi) \\ -sin(\varphi) & cos(\varphi) \end{bmatrix} \tag{10}$$

Da vi nu er i 3d indføres endnu en dimensions koordinater:

$$R_\varphi = \begin{bmatrix} cos(\varphi) & sin(\varphi) & 0 \\ -sin(\varphi) & cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

Denne matrix beskriver en cirkel der roteres om z-aksen, og dermed danner en kugle i (y,x)-planen. Vi vil beskrive en kugle i (x,z)-planet i (x, y, z)-koordinater, altså kartesiske koordinater, og derfor er det nødvendigt at bytte rundt på x- og y- koordinaterne. Det gør vi både for vores rotationsmatrix, og vektoren $r \cdot \begin{bmatrix} sin(\theta) \\ 0 \\ cos(\theta) \end{bmatrix}$, så vi får

$$\begin{bmatrix} -sin(\varphi & cos(\varphi) & 0 \\ cos(\varphi) & sin(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \left(r \begin{bmatrix} 0 \\ sin(\theta) \\ cos(\theta) \end{bmatrix} \right) = \begin{bmatrix} r \cdot sin(\theta) \cdot cos(\varphi) \\ r \cdot sin(\theta) \cdot sin(\varphi) \\ r \cdot cos(\theta) \end{bmatrix} \tag{12}$$

Når solens afstand estimeres til at være $r_s = 147.000.000.000m$, solens azimut-vinkel beskrives ved ϕ_s og dens zenit-vinkel beskrives ved θ_s, kan dens position altså beskrives med kartesiske koordinater på følgende måde:

$$\begin{bmatrix} r_s \cdot sin(\theta_s) \cdot cos(\phi_s) \\ r_s \cdot sin(\theta_s) \cdot sin(\phi_s) \\ r_s \cdot cos(\theta_s) \end{bmatrix} \tag{13}$$

Solpanelet placeres i origo. Det betyder, at panelets enhedsnormalvektor $\mathbf{u}_p$ har zenit-vinklen θ_p og azimut-vinklen ϕ_p. Når alle objekter placeres på himmelsfæren, kan vi ignorere radiussen. Enhedsnormalvektoren bliver så:

$$\mathbf{u}_p = \begin{bmatrix} sin(\theta_p) \cdot cos(\phi_p) \\ sin(\theta_p) \cdot sin(\phi_p) \\ cos(\theta_p) \end{bmatrix} \tag{14}$$

Samtidig er den normaliserede enhedsvektor for solen givet ved følgende sfæriske koordinater:

$$\mathbf{u}_s = \begin{bmatrix} sin(\theta_s) \cdot cos(\phi_s) \\ sin(\theta_s) \cdot sin(\phi_s) \\ cos(\theta_s) \end{bmatrix} \tag{15}$$

Vi kan regne $\langle \mathbf{u}_s, \mathbf{u}_p \rangle$ som prikproduktet mellem $\mathbf{u}_s$ og $\mathbf{u}_p$'s koordinater:

$$\mathbf{u}_s \cdot \mathbf{u}_p = \begin{bmatrix} sin(\theta_s) \cdot cos(\phi_s) \\ sin(\theta_s) \cdot sin(\phi_s) \\ cos(\theta_s) \end{bmatrix} \cdot \begin{bmatrix} sin(\theta_p) \cdot cos(\phi_p) \\ sin(\theta_p) \cdot sin(\phi_p) \\ cos(\theta_p) \end{bmatrix} \tag{16}$$

$$= sin(\theta_s) \cdot cos(\phi_s) \cdot sin(\theta_p) \cdot cos(\phi_p) + sin(\theta_s) \cdot sin(\phi_s) \cdot sin(\theta_p) \cdot sin(\phi_p) + cos(\phi_s) \cdot cos(\phi_p) \tag{17}$$

$$= sin(\theta_p) \cdot sin(\theta_s) \cdot cos(\phi_p - \phi_s) + cos(\theta_p) \cdot cos(\theta_s) \tag{18}$$

Prikproduktet regner den andel af solens normalvektor, der rammer solpanelet. Ifølge Theorem 2.1.6[4] er den absolutte værdi af prikproduktet mellem to vektorer mindre eller lig med produktet af de to vektorers længde. Da solpanelets og solens normalvektorer har længden 1, kan prikproduktet mellem dem altså kun have en absolut være på mindre eller lig med $1 \cdot 1 = 1$. Det vil sige, at grænserne for prikproduktet er $-1 \leq \langle \mathbf{u}_s, \mathbf{u}_p \rangle \leq 1$.

Som tidligere forklaret, afspejler prikproduktet af $\mathbf{u}_s$ og $\mathbf{u}_p$ orienteringen mellem solens og solpanelets normalvektorer. Når dette prikprodukt er 1, indikerer det, at vektorerne er parallelle og peger i modsat retning, hvilket svarer til maksimal solindstråling. Hvis prikproduktet er 0, er vektorerne ortogonale, hvilket betyder, at der ingen solindstråling er. I tilfælde af et negativt prikprodukt er normalvektorerne ensrettede. Hvis prikproduktet er -1, er de to vektorer fuldstændig parallelle i samme retning, hvilket indikerer, at panelet vender direkte væk fra solen, men da der ikke kan være negativ flux sættes prikproduktet altså bare til 0, hvilket også er lig med en flux på 0.

Funktionen solar-panel-projection-day er udviklet til at beregne solens normalvektors projektion ind på solpanelets normalvektor vha. solens zenit-vinkel (θ_s) og azimut-vinkel (ϕ_s), og panelets zenit-vinkel (θ_p) og azimut-vinkel (ϕ_p). Ved at omregne de sfæriske koordinater til enhedsvektorer, kan funktionen beregne prikproduktet af solens og panelets normalvektorer. Resultatet af dette prikprodukt er cosinus til vinklen mellem de to vektorer, hvilket direkte fortæller os, hvordan solpanelet er placeret i forhold til solens stråler. En vinkel på 0 grader mellem de to normalvektorer, hvor prikproduktet er 1, indikerer, at solpanelet er i den ideelle position, da det er vinkelret på solens stråler og modtager maksimal indstråling fra solen. Funktionen ses nedenfor:

```
def solar_panel_projection_day(theta_sol, phi_sol, theta_panel, phi_panel):
# Initialiser en liste til at gemme prikprodukterne for hver beregning
dot_products = []
for i in range(len(theta_sol)):
    #hvis solen er på den modsatte side af jorden:
    if np.deg2rad(theta_sol[i]) >= 0 and np.deg2rad(theta_sol[i]) <= np.pi/2:
        # Beregn enhedsvektorer for solens position og panelets position
        us = np.array([np.sin(theta_sol[i]) * np.cos(phi_sol[i]),
                       np.sin(theta_sol[i]) * np.sin(phi_sol[i]),
                       np.cos(theta_sol[i])])
```

```
            # Beregn enhedsvektoren for solpanelets normal ud fra de sfæriske koordinater
            up = np.array([np.sin(theta_panel[i]) * np.cos(phi_panel[i]),
                           np.sin(theta_panel[i]) * np.sin(phi_panel[i]),
                           np.cos(theta_panel[i])])

            # Beregn prikproduktet af de to enhedsvektorer, som repræsenterer vinklen mellem dem
            dot_product = np.dot(us, up)
        else:
            dot_product = 0
        # Prikproduktet må ikke være negativt; om nødvendigt sættes det til 0
        dot_product = max(0, dot_product)
        # Tilføj det beregnede prikprodukt til listen
        dot_products.append(dot_product)
        # Prikproduktet mellem solens og solpanelets enhedsvektorer udskrives
        print("Prikproduktet af us og up er:", dot_product)
    # Returner prikproduktet, hvis det er positivt, ellers returner 0
    return dot_products
```

Vi testede funktionen i tre forskellige situationer:

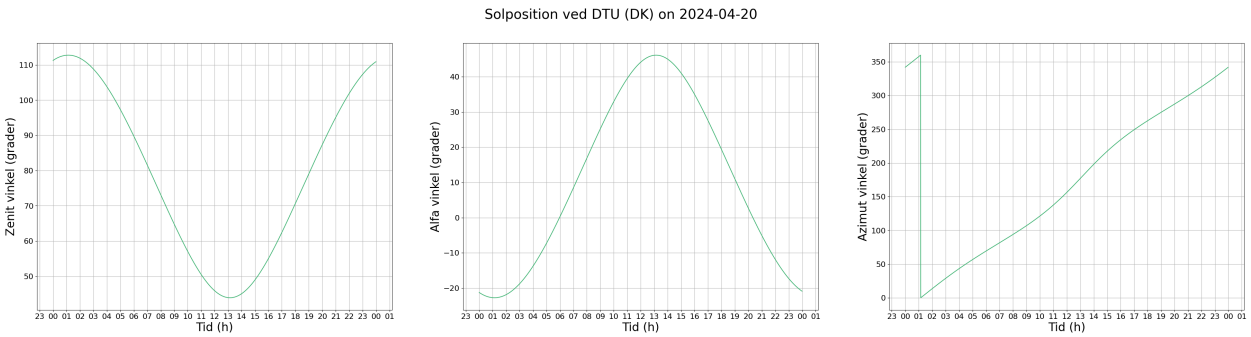
Situation 1: $\theta_s = \frac{\pi}{4}, \phi_s = \pi, \theta_p = 0, \phi_p = \pi$
I denne situation står solen stik syd, 45 grader over Jordens horisont. Samtidig står panelet lodret og vender også stik mod syd. Prikproduktets værdi er 0,707107, så ca 70% af solens indstråling rammer panelet.

Situation 2: $\theta_s = \frac{\pi}{2}, \phi_s = \frac{\pi}{2}, \theta_p = \frac{\pi}{2}, \phi_p = 0$
I denne situation står solen stik øst, 0 grader over Jordens horisont. Panelet ligger fladt ned på jorden. Ingen af solens irradians rammer derfor panelet, så derfor giver prikproduktet 0, med en lille fejl.

Situation 3: $\theta_s = 0, \phi_s = 0, \theta_p = \pi, \phi_p = 0$
I det tredje scenario befinder solen sig i zenit, dvs. højest på himlen. Panelet derimod er vendt 180 grader, så bagsiden af det vender op mod himlen. Dermed rammer ingen af solens irradians panelets forside, så prikproduktet bliver 0.

Solpositionsmodellering ved Pvlib

Som en del af simuleringen af solens bevægelse, har vi brugt Pvlib til at simulere solens zenit-, azimut- og alfa-vinkel, altså elevationsvinklen, hvert minut hele 2024. På Figur 3 nedenfor ses udviklingen af zenit-, elevations- og azimut-vinklen i løbet af den 20. april 2024.



Figur 3: Zenit-, elevations- og azimut-vinkel d. 20. april 2024

Plottet til venstre på figur 3 viser zenit-vinklen, altså hvor højt på himlen solen står. Det aflæses, at solen står højest ca. kl 13. Desuden kan følgende kode bruges til at beregne den højeste elevationsvinkel og størrelsen af denne:

```
max_elevation_time = solpos.loc[valgt_dato]['apparent_elevation'].idxmax()
max_elevation = solpos.loc[valgt_dato]['apparent_elevation'].max()
```

Dette giver samme resultat, netop at den højeste elevationsvinkel er 39°, som den når kl 13.00.

Når θ_s er over 90°, står solen under horisonten, og dermed er det mellem solnedgang og solopgang, altså nat. Da solens alfa vinkel, α_s, også måler solens position langs horisonten, og som nævnt, blot er zenit-vinklens komplementærvinkel, indikerer en negativ værdi altså også, at solen er "under"horisonten. Derved indikerer $\theta_s > 90°$ og $\alpha_s < 0°$, at det er nat.

Fra projektbeskrivelsen er det givet, at solen er står højest når zenit-vinklen er lig 0. Derved er solen på sit laveste når zenit-vinklen er 90°, og dermed er solopgang og solnedgang når zenit-vinklen er 90°. I et pythonprogram finder vi de steder, hvor zenit-vinklen netop er 90°. Vi finder da, at solopgangen er kl 5.53 og solnedgang er kl 20.26. Ifølge DMI er solopgang og solnedgang d. 20 april 2024 hhv. kl 5.52 og 20.27. Vores program er altså meget præcist.

Sommersolhverv finder sted d. 20. juni 2024. Solens vinkel findes som toppunktet for zenit-vinklens graf. Den findes ved følgende kode:

```
valgt_dato = "2024-06-20"
max_elevation_time = solpos.loc[valgt_dato]['apparent_elevation'].idxmax()
max_elevation = solpos.loc[valgt_dato]['apparent_elevation'].max()
```

Ifølge denne kode når solens sin højeste position kl. 13.12, hvor elevationsvinklen er $57,66°$. Nedenfor har vi ændret denne kode, så den finder solens højeste placering, α_{max} på en given dato og lokation.

```
def higest_sun(dato, lokation, tidszone):
    delta_tid = "Min"  # "Min", "H",

    site = Location(
        lokation[0], lokation[1], "tidszone"
```

```
    )  # breddegrad, længdegrad, tidszone, højde, navn

    # Definition af tidsintervaller for simulering
    times = pd.date_range(
        dato + " 00:00:00", dato + " 23:59:00", inclusive="left", freq=delta_tid, tz=tidszone
        )

    # Estimerer solposition med Lokationsobjektet
    solpos = site.get_solarposition(times)

    max_elevation_time = solpos.loc[valgt_dato]['elevation'].idxmax()
    max_elevation = solpos.loc[valgt_dato]['elevation'].max()
    return (max_elevation, max_elevation_time)
```

Denne funktion returnerer altså både α_{max} og hvornår på dagen denne nås.
Nedenfor har vi lavet et funktion, der omregner solens sfæriske koordinater i radianer til kartesiske koordinater. Vi estimerer stadig, at afstanden til solen, $r_s = 1,47 \cdot 10^{11}$ m . Vi bruger en parametrisering for en kugle i funktionen.

```
def spherical_to_xyz(theta, phi):
    x = r_s * np.sin(theta) * np.cos(phi)
    y = r_s * np.sin(theta) * np.sin(phi)
    z = r_s * np.cos(theta)
    return x, y, z
```

Til sidst i dette afsnit, har vi lavet et pythonprogram til omregning af solens position i kartesiske koordinater til sfæriske koordinater i radianer. Den første funktion udtrykker en værdi ved π. Den næste funktion omregner de kartesiske koordinater til sfæriske koordinater, som dermed bliver udtrykt ved π.

```
def float_to_pi_fraction(value):
    # Konverter et float til en string, der repræsenterer værdien som en brøk af pi
    fraction = Fraction(value).limit_denominator(12)
    if fraction.denominator == 1:
        if fraction.numerator == 1:
            return "pi"
        elif fraction.numerator == 0:
            return "0"
        else:
            return f"{fraction.numerator}pi"
    else:
        return f"{fraction.numerator}pi/{fraction.denominator}"

def xyz_to_spherical(x, y, z):
    r = np.sqrt(x**2 + y**2 + z**2)
    theta = np.where(r != 0, np.arccos(z / r) / np.pi, 0)
```

```
    phi = np.arctan2(y, x) / np.pi
    # Konverterer det numeriske resultat til en brøk af pi
    theta = np.vectorize(float_to_pi_fraction)(theta)
    phi = np.vectorize(float_to_pi_fraction)(phi)
    return r, theta, phi
```

Vi kan nu bruge disse omregninger fra kartesiske til sfæriske koordinater til at beregne fluxen af solens vektorfelt.

Effekt og energiberegning

Til at udregne fluxen af solens vektorfelt bruges udtrykket for flux gennem solpanelet, som vi definerede tidligere.

$$Flux = \langle u_s, u_p \rangle \cdot S_0 \cdot A_0 \cdot L \cdot B$$

I dataframet er en tabel med tidspunkter i minutter, azimut- og zenit-vinkler. Udtrykket ovenfor kan derfor bruges til at finde fluxen givet en azimut- og en zenit-vinkel på panelet. Funktionen returner en liste, der bruger den tidligere definerede funktion solar-panel-projection til at finde vinklen mellem solen og panelets normalvektor. Hvis zenit-vinklen er større end $\frac{\pi}{2}$, står solen under horisonten. Dermed sættes prikproduktet automatisk til at være 0, hvis $\theta_s > \frac{\pi}{2}$. Til sidst regnes fluxen i $\frac{W}{m^2}$ og afleveres i en liste med 1440 værdier, en for hvert minut.

```
def flux_day(valgt_dato, panelets_azimuth, panelets_zenith):
    s = 1100
    A0 = 0.5
    s0 = s*A0
    vinkel = solar_panel_projection((solposmin.loc[valgt_dato].zenith).to_numpy(),
    (solposmin.loc[valgt_dato].azimuth).to_numpy(), panelets_zenith, panelets_azimuth)
    return np.array(vinkel)*s0*L*B
valgt_dato = "2024-04-20"
print(flux_day(valgt_dato, np.pi, 0))
result_array = flux_day(valgt_dato, np.pi, 0)
print(result_array[719])
```

Indeks 720 svarer til kl. 12. Her returnerer funktionen 762W, hvilket vil sige at fluxen gennem solpanelet fra kl. 12.00 til 12.01 d. 24. april 2024 er 762W.

Vi kan nu bruge trapezmetoden til at integrere over denne liste af flux-værdier. På den måde findes energiproduktionen over en dag. Først defineres funktionen, og der tages højde for effektiviteten af solpanelet. Ifølge de angivne antagelser, kan effektiviteten af solpanelet defineres som $\frac{Wp}{L \cdot B}$ ved et flux på $1000\frac{W}{m^2}$. For at tilpasse energiproduktionen til den faktiske flux, som ikke altid er den ideelle $1000\frac{W}{m^2}$, kan vi korrigere beregningen ved først at dividere med denne standardiserede flux-værdi og derefter multiplicere med den aktuelle flux. På denne måde kan energiproduktionen beregnes for enhver given flux.

```
def trapez_day(valgt_dato, panelets_azimut, panelets_zenit):
    flux_val = flux_day(valgt_dato, panelets_azimut, panelets_zenit)
```

```
    value = 0
    efficiency = (425 / (L*B))/1000
    for i in range(1,len(flux_val)):
        value += ((flux_val[i-1] + flux_val[i]) / 2)*60*efficiency
    return value
```

Denne funktion returnerer en værdi for energiproduktionen. For d. 20. april 2024 er denne værdi 5.408.919 J for alle 24 timer samlet. Det svarer til 1,5 kWh.

I den følgende funktion finder vi energiproduktionen d. 20. april 2024 for hver heltalsvinkel. Panelets azimut-vinkel sættes til at være $\phi_p = 180°$, mens zenit-vinklen varierer.

```
valgt_dato = '2024-04-20'
energy_vinkel = [0 for i in range(91)]
for i in range(91):
    energy_vinkel[i] = (trapezt_day(valgt_dato, np.pi, i*(np.pi/180)))
```

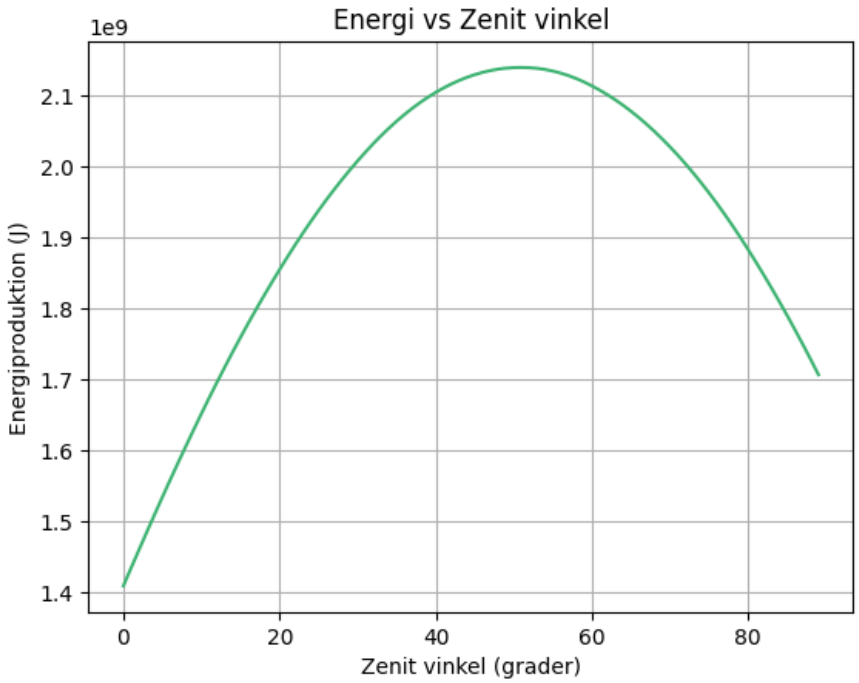
Vi finder, at energiproduktionen er højest, når solpanelets zenit-vinkel er 37°, hvor energiproduktionen er 110.590 J eller 111 kJ. En zenit-vinkel på 37° svarer til en elevationsvinkel på 53°. Dette er noget højere end 38°, som solpaneler normalt opstilles ved i Danmark. Det giver dog mening, at solpanelet optimalt skal være opstillet med en højere vinkel i april, da solen her ikke står lige så højt på himlen som i fx. juni. Dermed vil mere af solindstrålingen altså ramme et panel med en højere vinkel. I Tabel 1 nedenfor ses energiproduktionen (målt i kJ) for tolv vinkler mellem 0° og 90°. Det ses, at energiproduktionen stiger mellem 0° og 35° og derefter begynder at falde igen. Det stemmer godt overens med, at den optimale vinkel d. 20. april 2024 er $\theta_p = 37°$

θ_p	Energiproduktion (kJ)
0	5409
10	5916
20	6367
30	6591
35	6632
40	6626
45	6572
50	6469
60	6125
70	5602
80	4994
90	4185

Tabel 1: Energiproduktion fordelt på heltalsvinkler (θ_p), d. 20. april

Optimal vinkel

For at beregne energiproduktionen over et helt år for alle heltalsvinkler θ_p mellem 0° og 90°, skal vi ændre de tidligere definerede funktioner, så de anvender det rigtige tidsinterval. Når vi har gjort det, får vi en liste med 90 forskellige energiproduktioner; en for hver vinkel. Sammenhængen mellem vinklen og energiproduktion ses på Figur 4 nedenfor:



Figur 4: Energiproduktion for hver vinkel

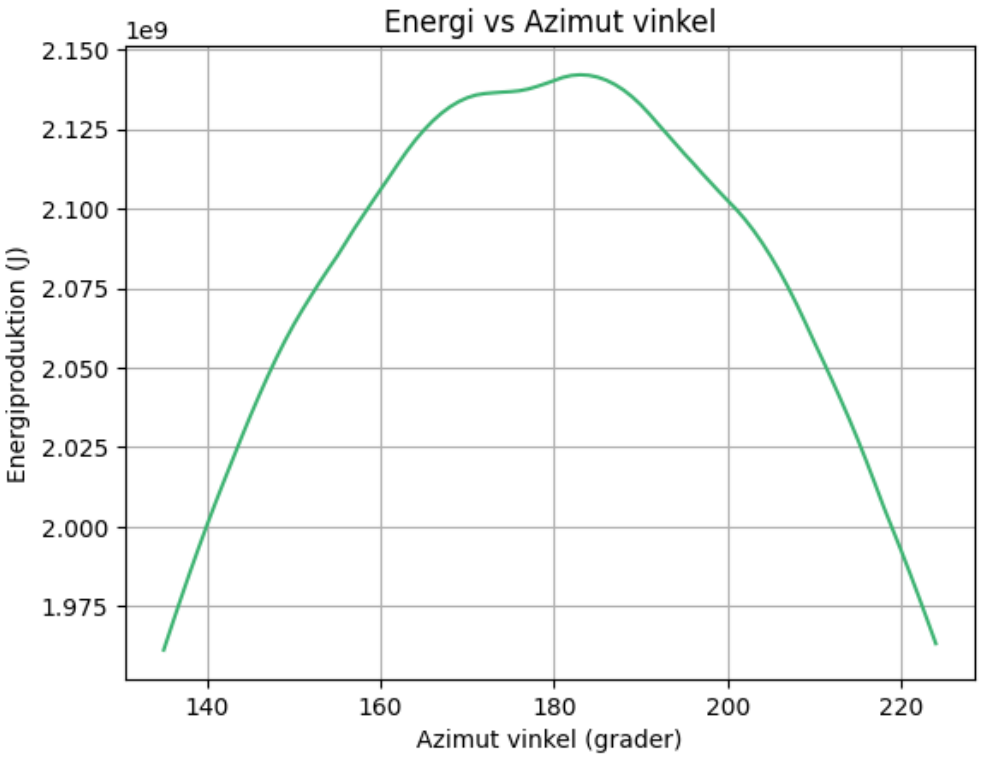
Vi har desuden udvalgt 11 vinkel-energiproduktion-par, som ses i Tabel 2 nedenfor:

θ_p	Energiproduktion (kJ)
0	1409840 kJ
10	1653966 kJ
20	1857926 kJ
30	2012211 kJ
40	2109578 kJ
50	2145109 kJ
55	2140523 kJ
60	2119830 kJ
70	2034095 kJ
80	1889432 kJ
90	1712968 kJ

Tabel 2: Energiproduktion fordelt på heltalsvinkler (θ_p), i løbet af hele 2024

Vi fandt, at den optimale vinkel er 51 zenit-grader, hvor energiproduktionen er 2.145.376 kJ, eller 595,9 kWh. Dette svarer til en elevationsvinkel på 39°. Normalt opstilles solpaneler i Danmark ved 38°. Vores program finder altså ca. den samme elevationsvinkel.

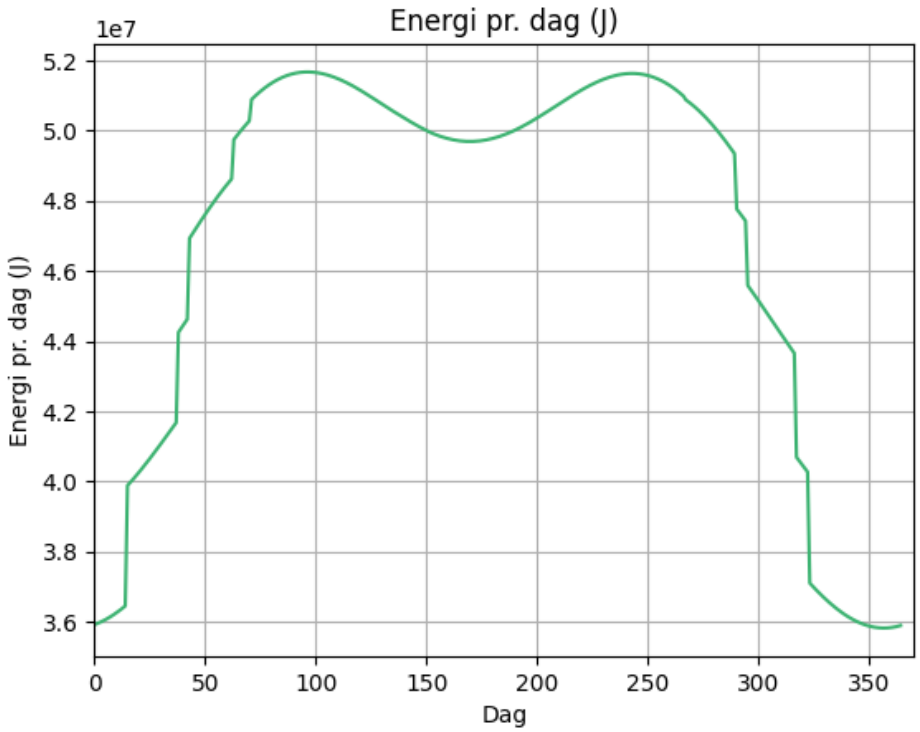
Efter at have fundet den optimale zenit-vinkel, ønsker vi at finde den optimale azimut-vinkel. Pga. solens bevægelse ift Danmark, regner vi med, at energiproduktionen er højest, når azimut-vinklen er mellem 135 og 225. Vi kører derfor et for-loop, der regner energiproduktionen for hver af disse vinkler, når zenit-vinklen er 51°. Sammenhængen mellem azimut-vinklen og energiproduktionen ses på Figur 4 nedenfor:



Figur 5: Energi plottet mod azimut-vinkel

Det ses på plottet, at energiproduktionen er størst, når azimut-vinklen er lige over 180°. Denne vinkel fandt vi til at være præcis 183°, hvor energiproduktionen er 2.147.228 kJ for et helt år. Dette svarer til 596,5 kWh. Den procentvise forskel mellem denne azimut-vinkel og en azimut-vinkel på 180° er $\frac{2147228kJ-2145376kJ}{2145376kJ} = 0,086\%$. Der er altså en meget lille procentvis afvigelse, når man bruger azimut-vinklen 183° ift. 180°. Til gengæld ses det på plottet, at hvis azimut-vinklen justeres meget, fx. så den er 140°, er energiproduktionen markant mindre.

Vi vælger at regne med otte solpaneler, der tilsammen udgør et areal på $1,134m \cdot 1,762m \cdot 8 = 16m^2$. På figur 5 nedenfor har vi plottet energiproduktionen pr. dag for de otte solpaneler, i løbet af hele 2024.



Figur 6: Energiproduktion pr. dag i løbet af 2024

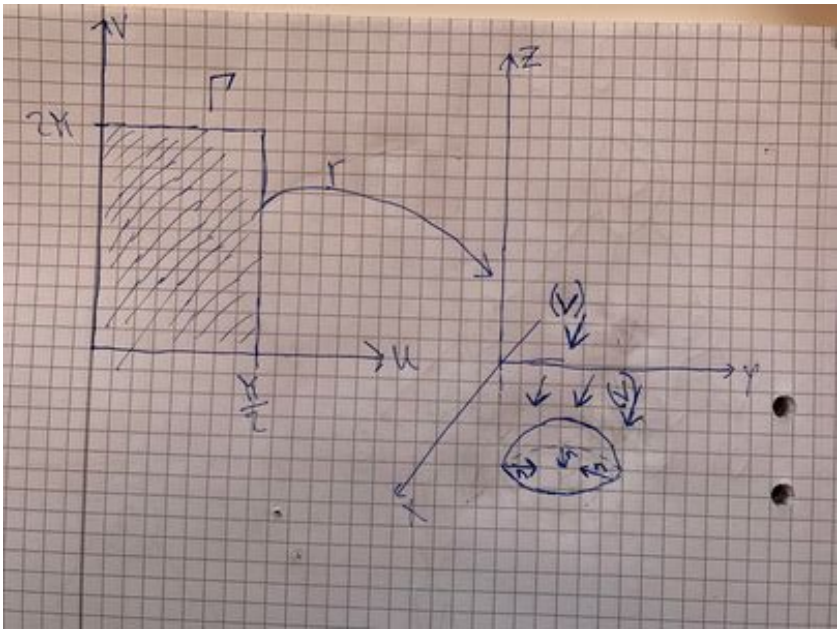
Det ses, at omkring dag 100, dvs. midt-april, og omkring dag 250, dvs. midt-august, er energiproduktionen højst. Indimellem falder energiproduktionen. Det betyder, at den største mængde af solens indstråling rammer panelerne i april og august. Indimellem står solen højere på himlen, og derfor er vinklen mellem de to normalvektorer større, og dermed er prikproduktet mindre, hvilket samlet giver en mindre energiproduktion.

Udvidelser

Krum flade

Vi kigger på lygtepæle på DTU, som tilnærmelsesvis er formet som en halvkugle. Her kan solpaneler påmonteres, som ikke ville genere studerende og personale, men vil kunne bidrage til den grønne omstilling.

Vi bruger sfæriske koordinater til at parametrisere en hul halvkugle. Forskellen er dog at en hul halvkugle har andre grænser.



Figur 7: Skitse over parametrisering af halvkugle

På figuren ovenover er skitseret parametriseringen af halvkuglen. Til venstre er parametriseringen r af halvkuglen, som er til højre. På halvkuglen er der indtegnet nogle af dens normalvektorer ($\mathbf{n}$), samt nogle pile, der repræsenterer solens vektorfelt ($\mathbf{V}$).

$$r(u,v) = \begin{bmatrix} a \cdot sin(u) \cdot cos(v) + p \\ a \cdot sin(u) \cdot sin(v) + q \\ a \cdot cos(u) + s \end{bmatrix}$$

I parametriseringen defineres halvkuglens radius som a, hvor vi tager udgangspunkt i en halvkugle med radius på 0.25 m. Zenit-vinklen betegnes med u og azimut-vinklen betegnes med v. Grænserne for disse vinkler er følgende:

$$0 \leq u \leq \tfrac{\pi}{2}$$
$$0 \leq v \leq 2\pi$$

Istedet for at u løber imellem 0 og π, som den ville gøre, hvis det var en kugle, løber den nu kun det halve, og dermed fås en halvkugle. Da a er fastsat, vil vi få en hul kugle, hvor afstanden er præcis a. I vores matematiske model repræsenterer parametrene p, q, og s koordinaterne for centrum af halvkuglen. På skitserne anvender vi en forenkling, hvor halvkuglens centrum er placeret i origo, det vil sige i punktet (0, 0, 0). Derfor sættes værdierne for p, q og s alle til nul. Dette resulterer i en mere direkte og simplificeret parametrisering af halvkuglen:

$$r(u,v) = \begin{bmatrix} 0.25 \cdot sin(u) \cdot cos(v) \\ 0.25 \cdot sin(u) \cdot sin(v) \\ 0.25 \cdot cos(u) \end{bmatrix}$$

Det indre af parametriseringen er injektiv, da ingen forskellige par af u og v, indeholdt i den indre mængde, giver det samme punkt, når det indsættes i parametriseringen r. Dette er dog ikke sandt for randen, hvor v lig 0 eller 2π med en given u værdi giver samme punkt ved brug af parametriseringen r. At randen er

injektiv er dog heller ikke et krav for en parametrisering. Parametriseringen er regulær da determinanten af Jacobi-matricen transponeret ganget med Jacobi-matricen er forskellig for nul på det indre, Definition 7.1.1.

$$det((J_r(u)^T J_r(u)) \neq 0$$

Jacobi-matricen, J_r er:

$$J_r = \begin{bmatrix} 0.25 \cdot cos(u) \cdot cos(v) & -0.25 \cdot sin(u) \cdot sin(v) \\ 0.25 \cdot sin(v) \cdot cos(u) & a \cdot sin(u) \cdot cos(v) \\ -0.25 \cdot sin(u) & 0 \end{bmatrix}$$

$$det(J_r(u)^T J_r(u)) = 0.25^4 \cdot sin^2(u)$$

Det ses, at determinanten kun afhænger af u, som løber imellem 0 og $\frac{\pi}{2}$. Det eneste tidspunkt udtrykket kan være 0, er når u er 0, men det er den kun på randen og derfor er det acceptabelt.

Udtrykket for fluxen for en given time kan nu findes ved hjælp af parametriseringen.

$$\int_F V \cdot n \, dS = \int_0^{\frac{\pi}{2}} \int_0^{2\pi} V(r(u,v)) \cdot (\tfrac{dr}{du} \times \tfrac{dr}{dv}) \, dvdu$$

Gøres dette for alle timer i løbet af et år, får vi følgende udtryk:

$$\sum_{n=1}^{365} (\sum_{i=1}^{24} (\int_0^{\frac{\pi}{2}} \int_0^{2 \cdot \pi} V(r(u,v) \cdot (\tfrac{dr}{du} \times \tfrac{dr}{dv}) \, dvdu))$$

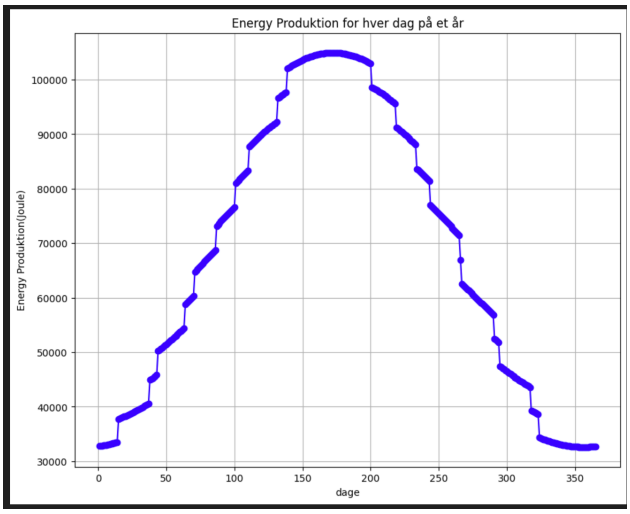
I summerne ovenover er n dage og i er timer.

Projektionen af solens vektorfelt ind på halvkuglens normalvektorer kan matematisk godt blive negativ, hvilket svarer til, at solen rammer bagsiden af panelet. I praksis giver dette dog ingen mening, så alle negative projektionsværdier sættes til 0. Vi har løst integralet/summen ved hjælp af numerisk integration og har regnet halvkuglens normalvektorer for hver femte grad. Vi projekterede solens vektorfelt ind på normalvektorene og summerede herefter over alle projektionsværdier og dividerede med antallet af normalvektorer. Dvs. vi fandt gennemsnitsværdien for projektionen. Herefter gangede vi med arealet af halvkuglen. Således fandt vi fluxen for en given time. Dernæst summerede vi over alle timer i 2024 (dvs. $24 \cdot 365 = 8760$). Vores valg af normalvektorer giver en hvis usikkerhed ift. præcisionen af den fundne energiproduktion. Havde vi arbejdet med flere normalvektorer havde vi haft langt flere beregninger og derfor en meget længere køretid, men til gengæld et mere præcist svar. Pythonprogrammet nedenfor udregner energiproduktionen for solpaneler på halvkugleformet lygtepæle.

```
#koden køres med dataframmen sat til timer.

#funktion til at lave en liste over krydsprodukteterne/normalvektorne af de
#afledte af parametriseringen ved at variere u og v
#up står for enhedsvektoren til panelet
def up_list():
    up_list_vals = []
    for i in range(0,91,5):
        for j in range(0,360,5):
```

```
            up = cross_product.subs({u:np.deg2rad(i),v:np.deg2rad(j), a:0.25})
            up_list_vals.append(up)
    return up_list_vals

#funktion til at bestemme prikproduktet mellem solens vektorfelt og panelets normalvektor
def solar_panel_projection_sphere(theta_sol, phi_sol, up_vals):
    dot_products = []
    dot_products_vinkel = []
    #omregner solens position fra grader til radianer
    theta_sol = (theta_sol*np.pi)/180
    phi_sol = (phi_sol*np.pi)/180
    #tjekker om solen er over horisonten
    if theta_sol >= 0 and theta_sol <= np.pi/2:

        #Beregner solens vektor udfra solens position
        us = np.array([np.sin(theta_sol) * np.cos(phi_sol),
                       np.sin(theta_sol) * np.sin(phi_sol),
                       np.cos(theta_sol)])

        #Beregner prikproduktet mellem solens vektor og panelets normalvektor
        for i in range(len(up_vals)):
                dot_product = np.dot(us, up_vals[i])
                #Prikproduktet må ikke være negativt, så derfor
                #sættes det til 0, hvis det er negativt
                dot_product = max(0, dot_product)
                dot_products_vinkel.append(dot_product)

        #finder summen af prikprodukterne for den time
        product_vinkel_sum = sum(dot_products_vinkel)
        dot_products.append(product_vinkel_sum)

    else:
        dot_product = 0
        dot_products.append(dot_product)

        # Returner en liste med prikprodukterne
    return dot_products

#funktion til at bestemme fluxen udfra prikproduktet
#mellem solens vektorfelt og panelets normalvektor
def flux_sphere():
    up_list_vals = up_list()
    projektioner = np.array([])
```

```
    s = 1100
    A0 = 0.5
    s0 = s*A0
    #laver et loop hvor den kalder sol_projektions funktionen for hver solposition på et år.
    for i in range(len(solpos.zenith)):
        projektioner = np.append(projektioner, solar_panel_projection_sphere((solpos.iloc[i].zenith),
        (solpos.iloc[i].azimuth), up_list_vals))
        #Her ganges projektionerne med overfladearealet af halvkuglen og
        #intensiteten af sollys og divideres med antallet af normalvektorer,
        #så vi får en gennemsnitlig værdi.
    return projektioner*(s0*4*np.pi*0.25**2*1/2)/((len(up_list_vals))

#funktion der ved brug af trapezmetoden bestemmer arealet
#under grafen af fluxen over et år, for vores halvkugle.
def trapez_sphere(valgt_dato):
    #kalder flux_sphere funktionen og gemmer dens output i flux_val
    flux_val = flux_sphere()
    value = []
    #de 212.7 er Wp/m^2 ligesom tidligere for solpanelet
    efficiency = (212.7 / (4*pi*0.25**2*1/2)) /1000
    count = 0
    summation = 0
    for i in range(1,len(flux_val)):
        summation += ((flux_val[i-1] + flux_val[i]) / 2)*60*efficiency*60
        #vi ganger med 60*60 for at få det i Joule, da vi har det i timer først
        count += 1
        print(count)
        if count == 24:
            print(summation)
            value.append(summation)
            summation = 0
            count = 0
    return value

#kalder trapez funktionen
sphere_energi = trapez_sphere(times)

#udregner hvor mange kWH halvkugle solcellen producerer per kvadratmeter på et år
(((np.sum(sphere_energi))/(4*np.pi*0.25**2*1/2))/3600000).evalf()
```



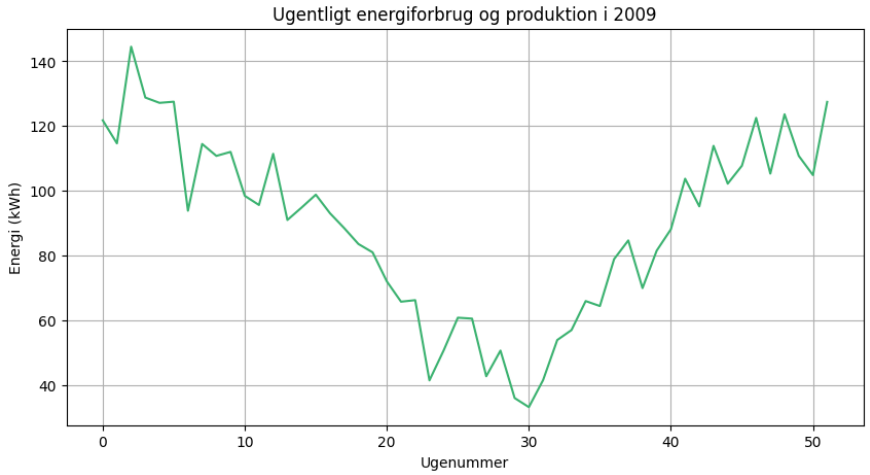
Figur 8: Energi produktion for et halvkugleformet solpanel med areal $0.39m^2$

Grafen herover viser energiproduktionen for et halvkugleformet solpanel i joule for hver dag over et hele 2024. Springene på grafen skyldes, at dataframen kører i timer frem for minutter. I den ovenstående kode kommer vi via numeriske integration frem til, at halvkugle-panelet på et år danner $24994,2kJ$ på dets $0.39m^2$, hvilket er svarende til $17,68kWh$ per kvadratmeter. I modsætning producerede det solpanel vi tog udgangspunkt i tidligere i opgaven ca. $297kWh$ per kvadratmeter, så halvkugle panelets energiproduktion per kvadratmeter er omtrent $\frac{1}{16}$ af energiproduktion fra det perfektvinklede solpanel. Det er altså evidens for, at et solpanel formet som en halvkugle ikke ville være at foretrække frem for et almindeligt firkantet, fladt panel. Den lave energiproduktion kan forklares ved, at størstedelen af det areal, hvor solpanelerne er monteret, ikke modtager optimal solindstråling i lange perioder af dagen. Desuden vil den del af panelet, der vender mod nord, næsten ikke producere strøm. Det er altså kun en meget lille del af halvkuglen, der er placeret i en optimal vinkel for solindfangning.

Klog energiproduktion

Tidligere i opgaven undersøgte vi hvilken kombinationen af azimut- og zenit-vinkler, der maksimerer den årlige energiproduktion. Dette er særligt fordelagtigt at optimere, hvis man har en stor batterikapacitet, der muliggør opbevaring af den producerede strøm til tidspunkter hvor solen ikke skinner. Det er imidlertid ikke alle, der har store batterier til at opbevare den genererede strøm. I sådanne tilfælde er det mere relevant at optimere vinklerne således, at de årlige energiomkostninger minimeres.

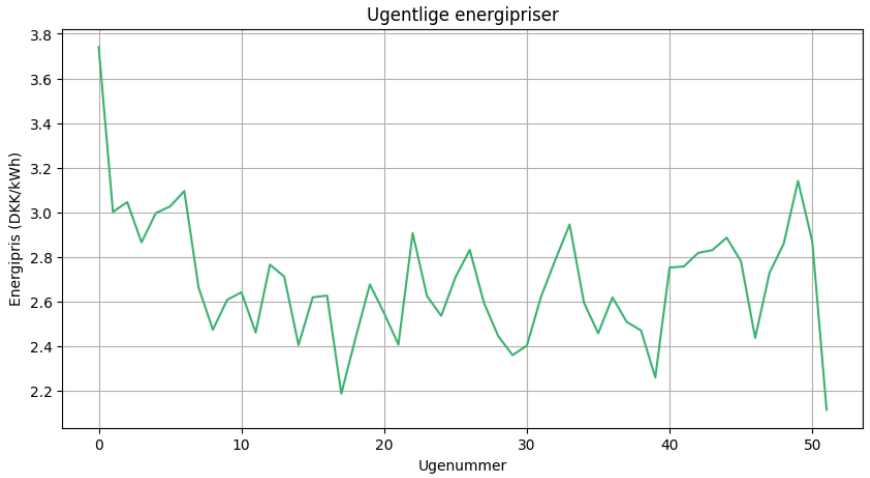
For at gøre dette er det nødvendigt at have adgang til data om energipriser i løbet af et år samt data om en gennemsnitlig husstands energiforbrug over samme periode. Vi fandt et datasæt med energipriser time for time fra 1950 til i dag. Til energiforbruget fandt vi et datasæt på Kaggle, som leverede minut-for-minut oplysninger om energiforbruget i en typisk husstand fra 2006 til 2010. Vi valgte at kigge på forbruget fra 2009, da dette var det eneste år uden manglende værdier. Derfor valgte vi også at bruge energipriserne i Danmark i 2009.



Figur 9: Energiforbrug i 2009

På figur 9 ovenfor ses grafen for det ugentlige energiforbrug i 2009. Det er tydeligt, at strømforbruget er højere om vinteren og lavere om sommeren. Allerede her overvejer vi derfor, hvilken betydning det har, når solpanelernes optimale vinkel skal bestemmes, med det mål at bruge færrest muligt penge på strøm.

Vi kommer i det følgende til at sammenligne energiproduktionen i 2009 med energiforbruget for samme år [5]. Vi kommer derefter til at optimere for elpriserne i 2009 [6]. Der er selvfølgelig en fejlkilde i form af finanskrisen, der påvirker vores data. Vores formodning er derudover at energiforbruget er betydeligt højere i 2024 end det var i 2009 og energipriserne er formodentlig anderledes i dag. For formålet med denne opgave er det heldigvis ikke særligt vigtigt. På figur 10 herunder er de ugentlige energipriser plottet.

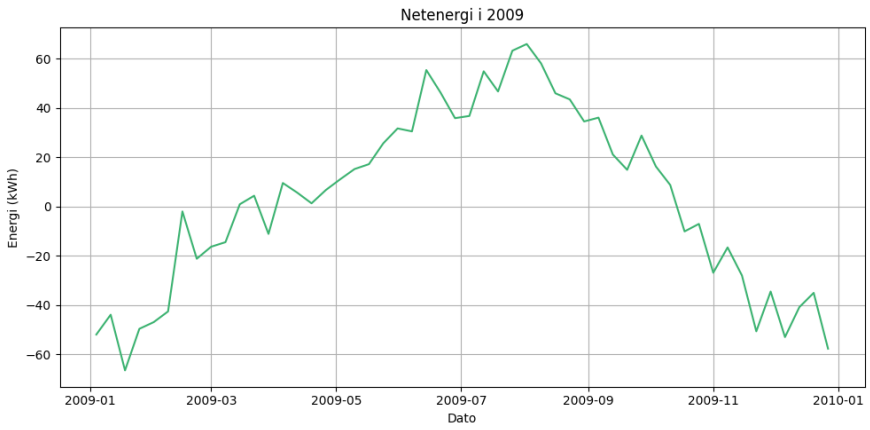


Figur 10: Ugentlige energipriser

Det ses, at energipriserne er højest i starten af året, og derefter generelt falder, inden det stiger igen om efteråret og vinteren. Dog svinger det rigtig meget fra uge til uge.

For at få et overblik over situationen simulerede vi produktionen med den optimale zenit- og azimut-vinkel,

vi fandt tidligere. Vi fandt derefter den totale energi, ved at trække energiforbruget fra energiproduktionen. På figur 11 nedenfor ses udviklingen i netenergi i løbet af 2009.



Figur 11: Netenergi

Vi ikke har kunnet finde noget brugbart data på salg af energi til el-nettet, og derfor antager vi, at det ikke er muligt at sælge overskydende energi. Det vil sige, at hver gang der er overskud af energi, sættes netenergien til 0. Hvis man ønsker at kunne sælge overskydende energi, skal dette selvfølgelig overvejes yderligere, inden solpanelerne opsættes.

Det er efterhånden åbenlyst, at optimering af energiproduktion generelt ikke nødvendigvis svarer til minimering af de årlige energiomkostninger. For at minimere energiomkostningerne, skal panelet vinkles sådan, at det i måderne med lav energiproduktion og høje energipriser producerer mest muligt energi, på bekostning af, at det producerer mindre energi i sommermånederne, hvor der alligevel er positiv netenergi. På den måde skal man købe mindre energi udefra i vintermåderne og dermed minimeres energiomkostningerne. For at gøre dette beregnede vi derfor energiproduktionen af otte solpaneler for hver kombination af zenit- og azimut-vinkler for hver time på et år. Fra denne trak vi den forbrugte strøm, hvorved vi fik netstrømmen og satte denne lig nul alle de steder den var større end nul. Herefter gangede vi med energiprisen for den pågældende time, og summerede de timevise energiomkostninger for at få de årlige energiomkostninger. Afslutningsvis fandt vi indexet med den laveste værdi, hvilket svarede til en bestemt vinkel.

Ved denne metode fandt vi, at en zenit-vinkel på 48 grader og en azimut-vinkel på 186 grader minimerer den årlige energiomkostning. Altså er zenit-vinklen en smule mindre, dvs. panelet står stejlere, end når man optimerer for maksimal energiproduktion. Samtidig er panelet en lille smule mere vest-vendt. Dette stemmer overens med det vi forventede. Solpanelerne skal stå lidt mere op, da mere af solen så rammer dem om vinteren, hvor priserne på energien er højest. Ligeledes er energien dyrest om aftenen ift. resten af dagen, og derfor giver det mening, at panelerne skal drejes mere mod venstre. Disse vinkler er dog ikke nødvendigvis de optimale i virkeligheden, da vi i vores model ikke tager højde for atmosfæren. Når solen står lavere på himlen, som den gør om vinteren og om aftenen, skal den passere mere atmosfære, hvilket reducerer intensiteten betydeligt. I vores model genererer solcellerne derfor mere strøm ved de store zenit-vinkler, end de ville i virkligheden, og det er derfor sandsynligt at den fundne zenit-vinkel er for lille og azimut-vinklen for stor i forhold til hvad der i virkeligheden er optimalt.

Konklusion

I dette projekt har vi undersøgt og analyseret de matematiske aspekter af solenergiproduktion ved hjælp af simulationsværktøjer og modelleringsteknikker. Vores primære fokus har været på effektiv anvendelse og optimering af solpanelers placering ift. solens position gennem året, for at maksimere energiudbyttet.

Først udviklede vi en funktion til beregning af solfluxen igennem et solpanel baseret på simuleringer af solens bevægelser, solpanelets geografiske lokation samt solpanelets zenit- og azimut-vinkler. Denne funktion bruges som fundament til at beregne de optimale vinkler. Vi fandt, at for et helt år, er den optimale zenit-vinkel for solpanelet 51 grader, mens den optimale azimut-vinkel er 183 grader. Dette indikerer, at for maksimalt energiudbytte bør solpanelet have en 39 graders vinkel med horisonten og pege mod syd og en lille smule mod vest. Ved denne opstilling fandt vi, at 8 solpaneler med et areal på i alt 16 m^2, ville have en energiproduktion på 596.5 kWh i løbet af hele 2024.

Dernæst analyserede vi effektiviteten af at montere et solpanel med en krum overflade på en gadelygte, og fandt herigennem ud af, at energiproduktionen for sådan et panel er omkring $\frac{1}{16}$ af energiproduktionen pr. kvadratmeter for et traditionelt fladt solpanel. Dette er altså markant lavere, og det ville derfor ikke være fordelagtigt at anvende solpaneler med en krum overflade på lygtepæle.

Endelig undersøgte vi, hvordan vinklen for otte solpaneler kunne optimeres baseret på forbrugsmønstre og energipriser med det formål at mindske energiomkostningerne. Vores analyse viser, at det er fordelagtigt at orientere vinklerne således, at panelet producerer forholdsvis mere strøm i de perioder, hvor energiforbruget er højt og energipriserne ligeledes høje, fx. vintermånederne og om aftenen. Til gengæld ville man da også minimere energiproduktionen i vintermånederne. Her er der dog positiv netenergi, så det øger ikke energiomkostningerne. Med dette mål fandt vi, at den optimale azimut-vinkel er 186° og den optimale zenit-vinkel er 48°. Med denne orientering maksimeres energiproduktionen om vinteren og om aftenen, hvor energipriserne er højere og resulterer i færre energiomkostninger. Hvorvidt dette i praksis vil være klogt, kræver en grundigere undersøgelse af atmosfærens påvirkning, for at bevise. Ville man udvikle videre på denne del, kun man have komplimenteret solpositionsmodellen, med historisk vejrdata for at optimere yderligere.

Vi har igennem matematiske modeller og simuleringer kunne identificere de mest effektive strategier for placering og orientering af solpaneler, taget i betragtning af solens bevægelse, den geografiske placering og økonomiske overvejelser. De traditionelle flade solpaneler, med en elevationsvinkel på 38 grader i Danmark er den mest effektive løsning, men vi kan konstatere, at den optimale konfiguration ikke er lige så simpel at finde frem til, og vil afhænge af præcis hvilket problem man ønsker at optimere.

Litteratur

[1] EnergiNet, "2022 sætter dansk rekord i vind og sol," 2022. [Online]. Available: https://via.ritzau.dk/pressemeddelelse/13667585/2022-saetter-dansk-rekord-i-vind-og-sol?publisherId=10304728

[2] S. H. Ipland, "Solenergi kan dække næsten 50% af verdens energiforbrug," 2020. [Online]. Available: https://illvid.dk/teknologi/energi/solceller/solenergi-kan-daekke-naesten-50-af-verdens-energiforbrug

[3] Viva-Energi, "Placering af anlæg," 2024. [Online]. Available: https://www.vivaenergi.dk/placering-solcelleanlaeg

[4] O. C. og Jakob Lemvig, "Mathematics 1b, functions of several variables," 2024. [Online]. Available: https://learn.inside.dtu.dk/d2l/lp/navbars/187824/customlinks/external/19616

[5] "Household energy consumption." [Online]. Available: https://www.kaggle.com/datasets/uciml/electric-power-consumption-data-set

[6] "Energipriser." [Online]. Available: https://www.energidataservice.dk/tso-electricity/Elspotprices?fbclid=IwAR0yqNxNpyoR_hr3ucBroIQC_z_9wAJ6K6gQLc7qH7lohc-ut8FYlVCk_H0