



Førsteårsprojekt 2023

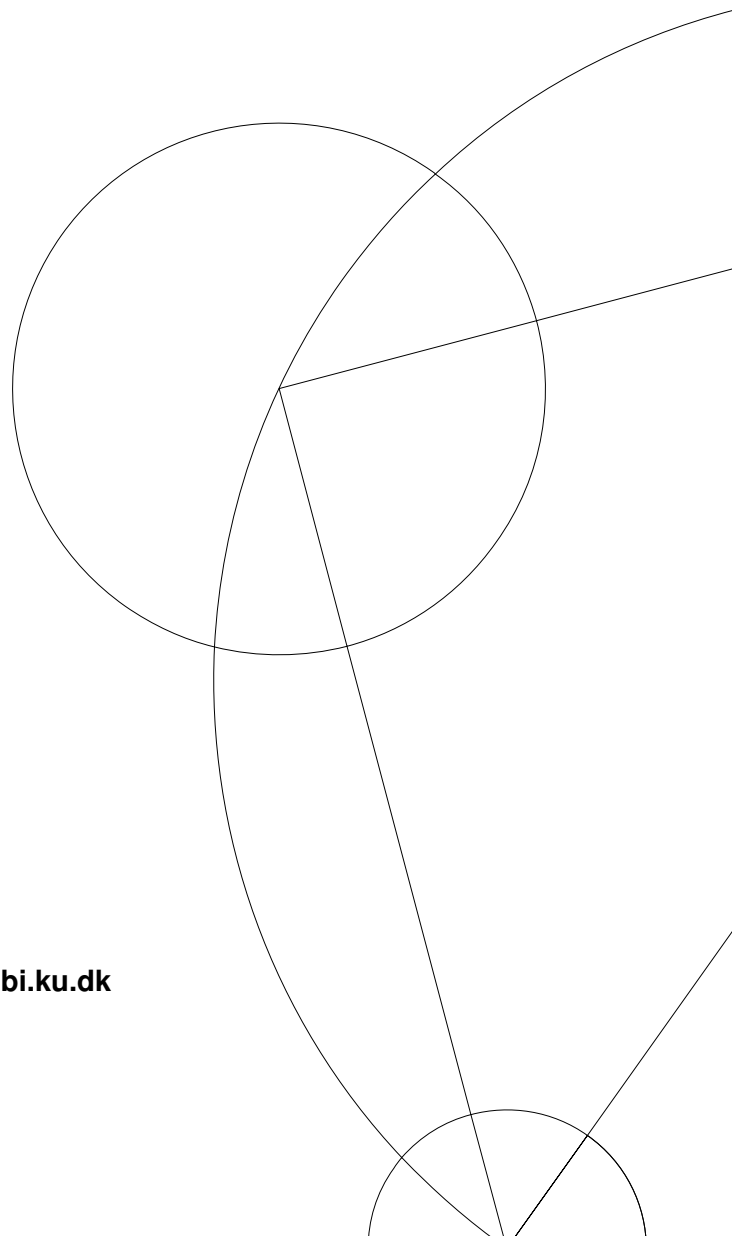
Projekts nummer: 2023-17

Projekt

Bliver solceller en kilde til global opvarmning?

Vejleder: Bo Møllersø Vinther Email: bvinther@nbi.ku.dk

Institut for Fysiske Fag. 14. marts 2025



Forfattere

Navn	KUId
Magnus Skjøtt Jensen	Klf637
Luc Holme Qvistgaard	Jqs943
Tobias Eichen	Xjr527

Rapporten omfatter 15 siders hovedtekst og 35 sides appendix

Rapporten er indsendt som en pdf-fil den 30. marts 2023

Resumé / Abstract

Dette projekt undersøger solcellers påvirkning på klimaet gennem en enkel energibalance-model, der er designet til at tilnærme oceanernes overfladetemperatur gennem simulation. Ved at observere Jordens energikredsløb kunne de vigtigste parametre, der påvirker Jordens energibalance – såsom strålingspåvirkningen af drivhusgasudledninger og Jordens overfladeegenskaber – identificeres og modelleres. Modelleringerne kunne herefter anvendes til at opstille en differentialligning, der beskriver ændringen af temperaturen i overfladen af Jordens oceaner. Ved hjælp af den numeriske metode Runge-Kutta af 4. orden og Monte Carlo kunne overfladetemperaturen af Jordens oceaner bestemmes med tilhørende usikkerheder i forskellige fremtidsscenarier. Dataene fra simulationen af de forskellige fremtidsscenarier viste, at i det værste scenarie skulle 16% af Sahara i år 2091 dækkes med solceller. Ved medregning af disse solcellers påvirkning på Jordens overfladeegenskaber ses det, at de kun vil bidrage til en temperaturstigning på $0.07, K$. Baseret på en kvalitativ sammenligning af rapportens temperaturmodel og modeller udgivet af IPCC vurderes de tilnærmede værdier at være tilfredsstillende nok til at konkludere, at solceller ikke vil være en væsentlig medvirkende faktor i klimakrisen.

This project will investigate the impact of solar panels on the climate through a simple energy balance model designed to approximate the surface temperature of the Earth's oceans through simulation. By observing the Earth's energy cycle, the most important parameters affecting the Earth's energy balance – such as the radiative forcing of greenhouse gas emissions and the Earth's surface properties – could be identified and modeled. These models could then be used to develop a differential equation describing the change in the surface temperature of the Earth's oceans. Using the 4th order Runge-Kutta numerical method and Monte Carlo, the surface temperature of the Earth's oceans could be determined, along with uncertainties in different future scenarios. Data from the simulation of various future scenarios showed that, in the worst-case scenario, 16% of the Sahara would be covered with solar panels in 2091. Taking into account the impact of these solar panels on the Earth's surface properties, it was found that they would only contribute to a temperature increase of about $0.07, K$. Based on a qualitative comparison between the temperature model in this report and models published by the IPCC, the estimated values are deemed satisfactory enough to conclude that solar panels will not be a significant contributing factor to the climate crisis.

Indhold

1	Introduktion	5
2	Teori	5
2.1	Drivhuseffekten	5
2.1.1	Indkommende kortbølget stråling	5
2.1.2	Solcellers indvirkning på albedoeffekten	6
2.1.3	Jordens langbølgede varmestråling	7
2.1.4	Jordens temperaturudvikling	8
2.2	Forskellige klimascenarier	8
3	Metode	9
3.1	Model	9
3.1.1	Numeriske løsning af differentialligninger	9
3.1.2	Runge-Kutta metoden af 4. orden	10
3.1.3	Havets totale varmekapacitet	11
3.1.4	Ændringen i havets totale energiindhold	12
3.2	Monte Carlo simulation	14
4	Resultater	15
5	Diskussion	17
6	Konklusion	19
7	Acknowledgements	19
8	Appendiks	22
8.1	Appendiks 1 - ligninger til beregning af drivhusgas-forcing	22
8.2	Appendiks 2 - Normalfordeling af parametre med usikkerheder	23
8.3	Appendiks 3 - Pythonkodens figurer	24
8.4	Appendiks 4 - Rå python kode	30

1 Introduktion

Mennesket har siden industrialiseringen udledt flere og flere drivhusgasser, og i takt med dette er der sket en temperaturenstigningen på jordoverfladen. Dette førsteårsprojekt beskæftiger sig med solceller og muligheden for, at teknologien bag kan blive en løsning på klimakrisen, eller om ændringen i jordens albedo på grund af de mørke solceller vil bidrage significant til opvarmningen. Vi vil undersøge forskellige fremtidige drivhusgasscenarier med og uden solceller og se, hvordan temperaturen udvikler sig for hvert af de givne tilfælde. Til at tjekke fremtidige scenarier er der bygget en model baseret på data fra foregående år. Ud fra tendenser i disse data vil vi arbejde med muligheden for at udarbejde en model over temperaturudviklingen for fremtiden. Med solceller som alternativ energikilde for de CO_2 emissionsrige kilder som kul og olie, er det også relevant at tjekke, om dette overordnet set kan lade sig gøre arealmæssigt.

2 Teori

2.1 Drivhuseffekten

Solen udsender kortbølget elektromagnetisk stråling. En del af den stråling, der rammer Jorden, vil blive absorberet og omdannet til termisk stråling eller energi. En vis procentdel af den stråling, Jorden modtager, vil dog blive reflekteret tilbage ud mod verdensrummet, hvor andelen, der reflekteres, kaldes albedoen.

Drivhuseffekten er et udtryk for, hvordan den manglende balance mellem Solens indstråling og Jordens udstråling forårsager en opvarmning af atmosfæren og jordoverfladen. Drivhuseffekten optræder naturligt på jorden og har spillet en essentiel rolle for klimaet lang tid før mennesket. Ubalancen, der er så meget snak om i medierne, skyldes, at mennesket leder massevis af drivhusgasser ud i atmosfæren, hvilket bidrager til en stærkere drivhuseffekt og derfor en stigning i temperatur.

2.1.1 Indkommende kortbølget stråling

Jorden fungerer som et sort legeme. Dette betyder, at den absorberer al den kortbølgede stråling fra Solen, som ikke reflekteres, mens den selv kun udsender en vis procentdel af denne energi som langbølget termisk stråling. For at beskrive denne indstråling fra Solen, benyttes en faktor for Solens intensitet i Jordens afstand fra Solen, der kaldes $I_{SW} = 1368 \frac{W}{m^2}$ [7]. Da

Jorden er sfærisk, rammer indstrålingen kun et tværsnit af Jorden, givet ved πR^2 , hvor R er Jordens radius på $6.371 \cdot 10^6 m$ [7]. Den sidste faktor, der dikterer den samlede effekt af den kortbølgede indstråling, er albedoen kaldt A . Albedoen er en faktor mellem nul og et, der beskriver hvor stor en procentdel af den kortbølgede indstråling, der reflekteres væk fra Jorden. Albedoeffekten beskriver helt generelt, hvordan lyse legemer reflekterer mere stråling end mørke legemer. Dette betyder altså, at albedoen af Jorden vil variere, alt efter hvilken side af Jorden der vender mod Solen, og hvilken årstid man befandt sig i. Albedoen beskrevet i ligning 1 forsimples altså til en gennemsnitsalbedo af hele Jorden over en længere årrække. Denne værdi kaldes, A_{jord} og approksimeres til 0.297 ± 0.005 [4]. Det resulterende udtryk for den kortbølgede stråling absorberet af Jorden er dermed:

$$P_{SW} = I_{SW}(1 - A)\pi R^2 \quad (1)$$

2.1.2 Solcellers indvirkning på albedoeffekten

Vores modellering af Jordens temperaturudvikling har til formål at medregne solcellers albedo, da det i modellen antages, at udviklingen af menneskehedens energibehov skal dækkes af solceller placeret i Sahara. Albedoen af Jorden med solceller, A_{jms} , kan deles op i tre komponenter, albedoen af den andel Sahara dækker af Jorden, albedoen af den andel de tilførte solceller dækker af jorden, og albedoen af den andel af Jorden, der hverken er dækket af solceller, eller af Sahara. Fra disse tre led kan et forenklet udtryk for A_{jms} udledes således:

$$\begin{aligned} A_{jord} &= A_{Sahara} \cdot f_{Sahara} + A_{ikke\ Sahara} \cdot (1 - f_{Sahara}) \approx 0.297 \pm 0.005 \\ A_{jms} &= A_{Sahara} \cdot (f_{Sahara} - f_{solceller}) + A_{solceller} \cdot f_{solceller} + A_{ikke\ Sahara} \cdot (1 - f_{Sahara}) \\ &= A_{Sahara} \cdot f_{Sahara} - A_{Sahara} \cdot f_{solceller} + A_{solceller} \cdot f_{solceller} + A_{ikke\ Sahara} \cdot (1 - f_{Sahara}) \\ &= A_{jord} + (A_{solceller} - A_{Sahara}) \cdot f_{solceller} \end{aligned} \quad (2)$$

Her betegner, $A_{solceller} = 0.1$ [10] og $A_{Sahara} = 0.4$ [10], henholdsvis albedokoefficienterne for solceller og Sahara. $f_{solceller}$ betegner den procentdel af Jorden der er dækket af solceller placeret i Sahara. f_{Sahara} betegner den procentdel Sahara udgør af Jorden, og $A_{ikke\ Sahara}$ betegner albedoen af Jorden uden Sahara. Disse to værdier er underordnende, da de forsvinder i udledningen. I vores model kommer værdien for $f_{solceller}$ til at variere med tiden, da menneskehedens energibehov kommer til at stige med årene. Til at beregne $f_{solceller}$ som funktion af tiden, benyttes flere parametre. Først udregnes den årlige energi produceret per m^2 solcelle i Sahara, kaldt E_{ps} . Til dette benyttes, I_{SW} , antal timer på et år kaldt, H_{py} ,

og solcellers effektivitet, η , der er antaget til at være 13.5% effektive[10]. Udtrykket for E_{ps} bliver:

$$E_{ps} = \frac{H_{py}(I_{SW} \cdot 0.25)\eta}{1000} \quad (3)$$

Hvor det noteres at I_{SW} kun antages en fjerdedel effektiv, da Solen ikke altid er fremme i Sahara eller står vinkelret på solcellerne. Udtrykket deles med 1000 for at omregne fra watt timer til kilowatt timer. En anden parameter, der benyttes, er en funktion for hvor meget energi, der skal erstattes af solceller, kaldt $E_s(t)$. Det er denne værdi, der kommer til at variere med tiden. Med værdien for Jordens totale overfladeareal, $Areal_{Jord} = 5.101 \cdot 10^8 km^2$ [10], og udtrykket for overfladearealet dækket af solceller, $Areal_{solceller}(t) = \frac{E_s(t)}{E_{ps}}$, kan et udtryk for $f_{solceller}(t)$ skrives:

$$f_{solceller}(t) = \frac{Areal_{solceller}(t)}{Areal_{Jord}} \quad (4)$$

2.1.3 Jordens langbølgede varmemstråling

Givet Jordens opførelse som et ikke perfekt sortlegeme, udsender den som beskrevet tidligere også langbølget termisk stråling. Intensiteten af denne udgående stråling er givet ved Stefan-Boltzmanns lov: $I_{LW} = \sigma T^4$. Hvor T beskriver sortlegemets temperatur, og σ er Stefan-Boltzmanns konstant, der er $5.67 \cdot 10^{-8} \frac{W}{m^2 K^4}$ [7]. Eftersom Jorden udsender denne termiske stråling, er det hele Jordens overfladeareal på $4\pi R^2$ der bruges til at beregne den samlede effekt af den langbølgede udstråling.

Jordens udstråling beskrives yderligere af to faktorer, emissiviteten ϵ og drivhusgasforcingen ΔF . Emissiviteten angiver procentdelen af energi, der ikke fanges i atmosfæren, den er altså et udtryk for, hvor tæt et objekt er på et perfekt sortlegeme. I Jordens tilfælde er emissiviteten temperaturafhængig, hvilket betyder, at emissiviteten aftager, når Jorden varmes op. Dette skyldes at en opvarmning af Jorden resulterer i øget vanddamp, der gør emissiviteten lavere. Drivhusgasforcing beskriver den ekstra varme, som fastholdes i atmosfæren på grund af de menneskeskabte udledte drivhusgasser såsom kuldioxid, metan og kvælstof. Disse gasser interagerer i større grad med den langbølgede infrarøde udstråling fra Jorden end med den kortbølgede indstråling fra Solen og er på den måde i stand til at fange varmen i atmosfæren. Udtrykket for den samlede effekt af den langbølgede udstråling fra Jorden bliver således:

$$P_{LW} = (\epsilon\sigma T^4 - \Delta F)4\pi R^2 \quad (5)$$

Drivhusgasforcingen er den faktor, der forårsager uligheden i Solens indstråling og

Jordens udstråling, hvilket leder til Jordens observerede temperaturstigninger.

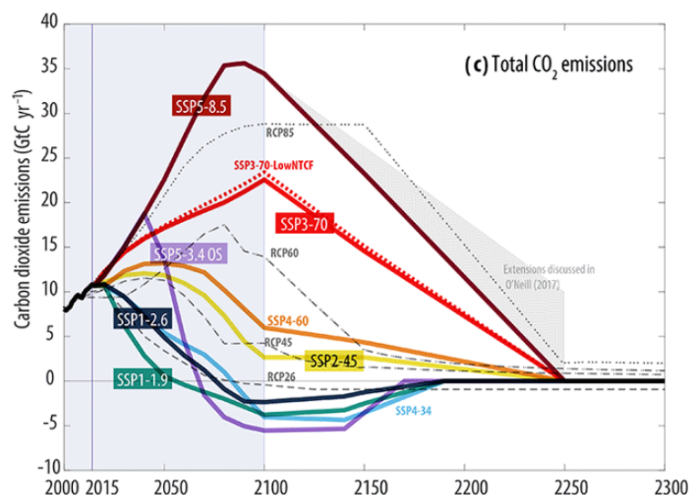
2.1.4 Jordens temperaturudvikling

Som nævnt tidligere stammer Jordens temperaturudvikling fra den manglende balance mellem Solens kortbølgede indstråling og Jordens langbølgede udstråling. I vores model antager vi, at al den energi, der bliver på Jorden, ender med at blive optaget i Jordens oceaner, da oceanerne næsten absorberer al stråling, der rammer dem grundet deres dybde og lave albedoværdi. Denne antagelse gør det muligt for os at opstille en differentialligning til at beskrive temperaturudviklingen i Jordens øvre havlag, ved hjælp af ligning 1, ligning 5 og havets totale varmekapacitet kaldt, C_{ocean} :

$$\frac{d}{dt}T_{ocean}(t) = \frac{P_{SW}(t) - P_{LW}(t, T_{ocean}(t))}{C_{ocean}} \quad (6)$$

2.2 Forskellige klimascenarier

Jordens fremtidige temperatur er afhængig af, hvor mange drivhusgasser vi udleder. I artiklen, *The shared socio-economic pathway (SSP) greenhouse gas concentrations and their extensions to 2500*, af Meinshausen et al[11] bliver der præsenteret fem forskellige klimascenarier, som de har modelleret.



Figur 1: De fem forskellige SSP scenariers forudsigelser af CO_2 emission per år.[11]

De fem forskellige scenarier, SSP1-1.9, SSP1-2.6, SSP2-4.5, SSP3-7.0 og SSP5-8.5, beskriver alle forskellige fremtidsscenarier for udvikling af CO_2 emission på Jorden. Scenarie SSP1-1.9, beskriver en fremtid hvor Jordens CO_2 emission stoppes i cirka år 2050 (se figur 1), da der ikke bruges flere fossile brændsler. Scenariet beskriver ligefrem, hvordan CO_2

emissionen omvendes ved at hive CO_2 ud af atmosfæren og ned i jorden. Det er også dette scenarie, der vil sørge for, at vi når i mål med Parisaftalen, hvis mål er at gennemsnitstemperaturen ikke stiger med mere end $1.5^\circ C$ sammenlignet med før industrialiseringen. Scenarie SSP1-2.6 beskriver en fremtid hvor vi stopper med at udlede CO_2 i år 2070 (se figur 1), men dog ikke ligeså hurtigt som i SSP1-1.9. I scenarie SSP2-4.5 vil CO_2 emissionen stige kortvarigt og herefter aftage langsomt. I SSP3-7.0 scenariet vil CO_2 emissionen stige markant, da brugen af fossile brændstoffer vil øges, og de CO_2 neutrale energiformer vil blive nedprioriteret. I scenarie SSP5-8.5, vil CO_2 emissionen stige mest. Her vil man dække al energi med fossile brændsler, og man vil ikke skifte til CO_2 neutrale løsninger. Disse forskellige scenarier vil medføre forskellige temperaturstigninger [11] grundet deres forskellige drivhusgasemissioner.

3 Metode

3.1 Model

Som beskrevet tidligere kan temperaturudviklingen i de øvre lag af Jordens oceaner, beskrives med differentialligningen fra ligning 6. Eftersom denne differentialligning er relativt kompleks, kan den ikke løses analytisk, derfor tilnærmes temperaturudviklingen numerisk.

3.1.1 Numeriske løsning af differentialligninger

Den numeriske metode er kort sagt en måde at løse matematiske og fysiske ligninger ved hjælp af diskrete udregninger. Ved at anvende den kontinuerte differentialligning i enkelte diskrete punkter, kan en ændring af temperaturen til et bestemt tidspunkt beregnes, og derefter bruges til at finde temperaturen, et lille tidsskridt, dt , frem i tiden. Denne metode bygger på, at udviklingen af temperaturen antages lineært, i det korte tidsrum dt . Den numeriske løsningsmetode af ligning 6 kommer dog med begrænset præcision, da den benytter sig af en diskret fremstilling, der ultimativt kun er en tilnærmelse af den reelle udvikling. For at minimere dette tab af præcision, vælges normalt en meget lav værdi af dt . Dette medfører dog, at den påkrævede computerkraft bliver meget større, da den er omvendt proportional med størrelsen på dt .

I modellen til beregning af temperaturudviklingen i de øvre lag af Jordens oceaner, benytter vi Monte Carlo til beregning af usikkerheder på temperaturen. Benyttelsen af

Monte Carlo forklares og diskuteres senere, men metoden resulterer ultimativt i en faktor 100 større behovet computerkraft end originalt.

3.1.2 Runge-Kutta metoden af 4. orden

Givet den forøgede nødvendige mængde computerkraft, vil en numerisk tilnærmelse af temperaturudviklingen ved hjælp af Eulers metode, den enkleste numeriske løsnings metode, være ineffektiv. Det er derfor, den mere effektive Runge-Kutta metode af 4. orden benyttes til at løse ligning 6 numerisk. Runge-Kutta metoden af 4. orden bygger på samme grundprincipper som Eulermetoden. Hvis $\frac{dy}{dt} = f(t, y)$, kan et tidsskridt frem beskrives således:

$$\begin{aligned} k_1 &= dt \cdot f(t_n, y_n) \\ y_{n+1} &= y_n + k_1 \end{aligned} \tag{7}$$

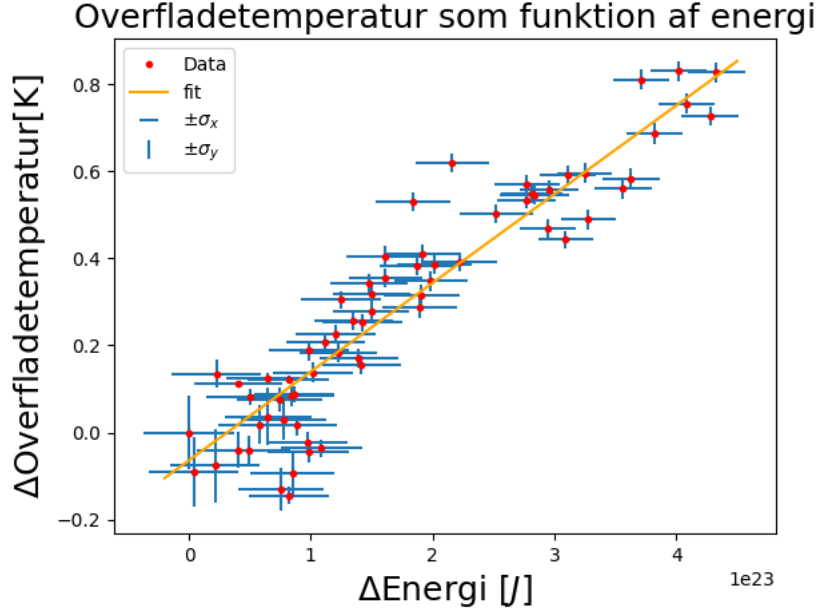
Denne fremskrivning af y -værdien, kaldes en Eulerfremskrivning, hvor k_1 kan anses som ændringen i den approksimerede funktionsværdi, kaldt fremskrivningsværdien. I Runge-Kutta metoden benyttes denne Eulerfremskrivning, til at tage et testskridt frem til midterpunktet af intervallet, hvorfra t og y værdierne her kan benyttes til at beregne en mere eksakt værdi af $f(t, y)$ der så kan bruges til at tage et tidsskridt frem. Runge-Kutta metode af 4. orden, benytter i alt fire af disse testskridt, hvortil et vægtet gennemsnit af de fire fremskrivningsværdier, vil give en væsentligt mere præcis approksimeret funktionsværdi. Denne metode er beskrevet således:

$$\begin{aligned} k_1 &= dt \cdot f(t_n, y_n) \\ k_2 &= dt \cdot f\left(t_n + \frac{1}{2}dt, y_n + \frac{1}{2}k_1\right) \\ k_3 &= dt \cdot f\left(t_n + \frac{1}{2}dt, y_n + \frac{1}{2}k_2\right) \\ k_4 &= dt \cdot f(t_n + dt, y_n + k_3) \\ k_g &= \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\ y_{n+1} &= y_n + k_g \end{aligned} \tag{8}$$

Hvor k_g er det vægtede gennemsnit af de fire fremskrivningsværdier. Denne metode medfører at vi går fra en afvigelse, fra den reelle temperatur, der afhænger af dt^1 , til en afvigelse der afhænger af dt^4 [12]. På denne måde kan vores model altså køre med større tidsskridt, og derved sparre computerkraft, men stadig vedligeholde samme præcision, så længe vi benytter et dt mindre end et.

3.1.3 Havets totale varmekapacitet

Varmekapaciteten af havet antages at være konstant i vores model. Den totale varmekapacitet af havet kan beregnes fra to værdier, den totale mængde energi i havet og den gennemsnitlige overfladetemperatur af havet. Selvom man egentligt burde anvende oceantes gennemsnitstemperatur i sådan en beregning, så er det overfladetemperaturen, der bestemmer udstrålingen fra havet og derfor den, der er interessant.



Figur 2: Lineært fit på data for udviklingen af havets gennemsnitlige temperatur fra 1955 til 2020[9], som funktion af havets udvikling af energiindhold fra 1955 til 2020[6]. $\chi^2_{red} = 2.56$.

Som set på figur 2, kan en lineær model, $y = ax + b$, fittes til vores data, hvortil C_{ocean}^{-1} kan findes som fitparameteren a , der bestemmes ved hjælp af et curvefit bygget på the least squares method[2]. C_{ocean}^{-1} og dens tilhørende usikkerhed, bestemmes i stedet for C_{ocean} , da vi gerne vil forsikre at usikkerhederne på overfladetemperatur ændringerne, er dem parametre der vægtes højest, dog medregnes energiændringens usikkerhedsbidrag på overfladetemperatur ændringernes usikkerheder også. Usikkerheden på C_{ocean}^{-1} kan findes fra formelen[2]:

$$\sigma_m = \sqrt{\frac{\sigma^2}{N(\overline{x^2} - \bar{x}^2)}} \quad (9)$$

Hvor $\sigma = \sqrt{\Delta_y^2 + \Delta_{y, fra x}^2}$ [2], og N er antal datapunkter. Fra de benyttede data udregnes det at $C_{ocean}^{-1} = (2.04 \pm 0.06) \cdot 10^{-24} \frac{K}{J}$, hvor det tilpassede fit har et reduceret chi-kvadrat på 2.56. Dette lave reducerede chi-kvadrat bekræfter, at dataens usikkerheder er blevet behandlet ordenligt, og antagelsen om den lineære tendens i dataen, virker passende. Dette

bekræfter, at det er en ok tilnærmelse at beregne varmekapaciteten mht. overfladetemperaturen.

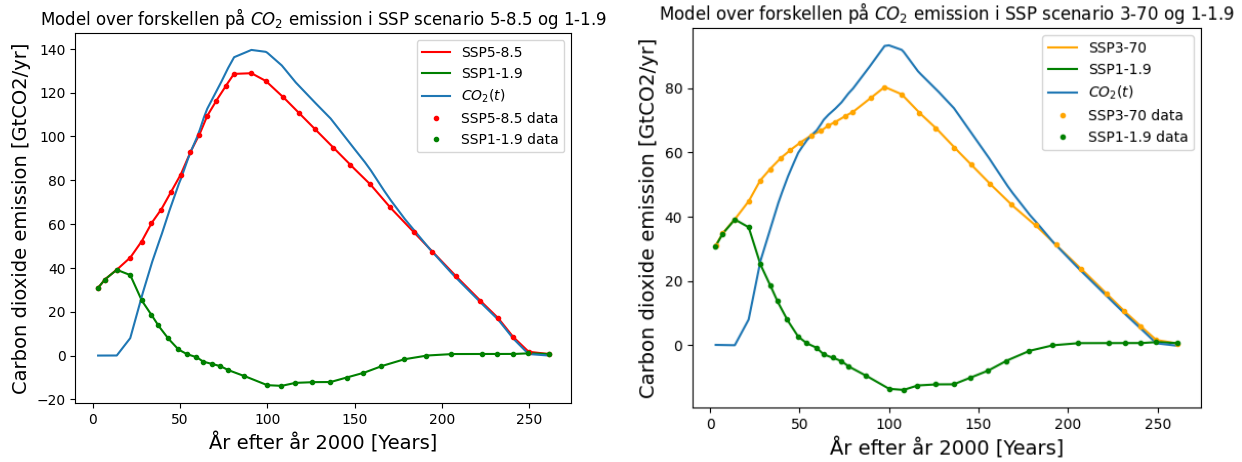
3.1.4 Ændringen i havets totale energiindhold

I ligning 6 angiver leDET, $P_{SW}(t) - P_{LW}(T_{ocean}(t), t)$, ændringen i havets totale energiindhold til tiden. Den kortbølgede indstråling fra ligning 1, benytter albedokoefficienten, A_{jms} der beskrives i kapitel 2.1.2. Denne værdi afhænger af tiden efter år 2015, da vi antager at vi i år 2015 begynder at dække det voksende energibehov med solceller. Som beskrevet i ligning 4 kræves en model for hvor meget energi der skal erstattes af solceller, altså $E_s(t)$, for at kunne bestemme $A_{jms}(t)$.

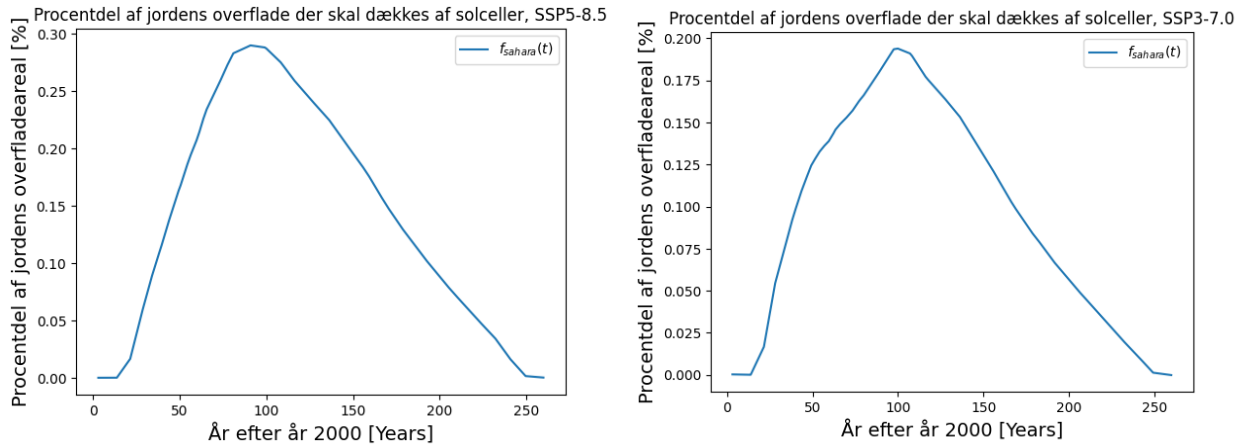
For at udregne hvor mange solceller der skal bruges til at dække fremtidens ekstra energibehov for henholdsvis scenarie SSP3-7.0 og SSP5-8.5, benytter vi figur 1 til at beregne hvor meget CO_2 emission der skal dækkes af solceller hvert år. Modellen bygger på at se på forskellen mellem CO_2 emission i SSP3-7.0/SSP5-8.5 og SSP1-1.9. I benyttelsen af denne forskel ligger en antagelse om, at menneskets forbrug og levevilkår vil fortsætte med at stige, men vi stadig følger det bedste scenarie for CO_2 emission, da forskellen mellem SSP1-1.9 og SSP3-7.0/SSP5-8.5 erstattes af bæredygtig solcelleenergi. Til approksimation af denne funktion for CO_2 emission der skal dækkes, bruges logger pro til at placere punkter langs de relevante scenarier på figur 1. Disse punkter benyttes til at finde funktioner, ved hjælp af interpolation, der passer til udviklingen af det givne SSP-scenariers CO_2 emission. Herefter har vi lavet funktioner der beskriver forskellen mellem de to polynomier, se figur 3. Denne forskel er mængden af CO_2 der skal erstattes af solceller. Denne mængde CO_2 emission kan direkte omregnes til en energimængde der skal erstattes, da der pr. kg CO_2 udledt, produceres $4.289kWh$ energi[3].

Med difference funktionen ved vi nu hvor meget CO_2 , og derved også hvor meget energi, solcellerne skal erstatte i fremtiden, $E_s(t)$. Ved brug af, E_{ps} fra ligning 3, kan arealet af de benyttede solceller, og derved også procentdelen de udgør af Jordens overfladeareal beregnes fra ligning 4.

Fra disse funktioner kan, $A_{jms}(t)$ approksimeres, og derved kan $P_{SW}(t)$ bestemmes fra ligning 1. Med effekten af Solens indkommende kortbølgede stråling bestemt, mangler kun effekten af Jordens langbølgede udkommende stråling for at kunne bestemme ændringen i havets totale energiindhold. Fra ligning 5 ses det, at de eneste uestimerede værdier er,



Figur 3: Modelleringer af to forskellige sceneriers mængde CO_2 der skal erstattes af solceller.



Figur 4: Figuren viser procentdelen af Jordens overfladeareal solcellerne skal dække, fra år 2000 og 250 år frem,

Jordens emissivitet, og drivhusgasforcingen. Bestemmelsen af Jordens emissivitet bygger på antagelsen af, at indstråling og udstråling fra Jorden er i ligevægt uden drivhusgasforcing. Ved denne antagelse kan vi sætte ligning 1 lig med ligning 5, og på den måde finde Jordens emissivitet ved, T_0 i år 1850.

$$I_{SW}(1 - A_{jord})\pi R^2 = \epsilon_0 \sigma T_0^4 4\pi R^2 \Rightarrow \epsilon_0 = \frac{I_{SW}(1 - A_{jord})}{\sigma T_0^4} = \frac{1368 \frac{W}{m^2} \cdot (1 - 0.297)}{5.67 \cdot 10^{-8} \frac{W}{m^2 K^4} \cdot (286.8 K)^4 \cdot 4}$$

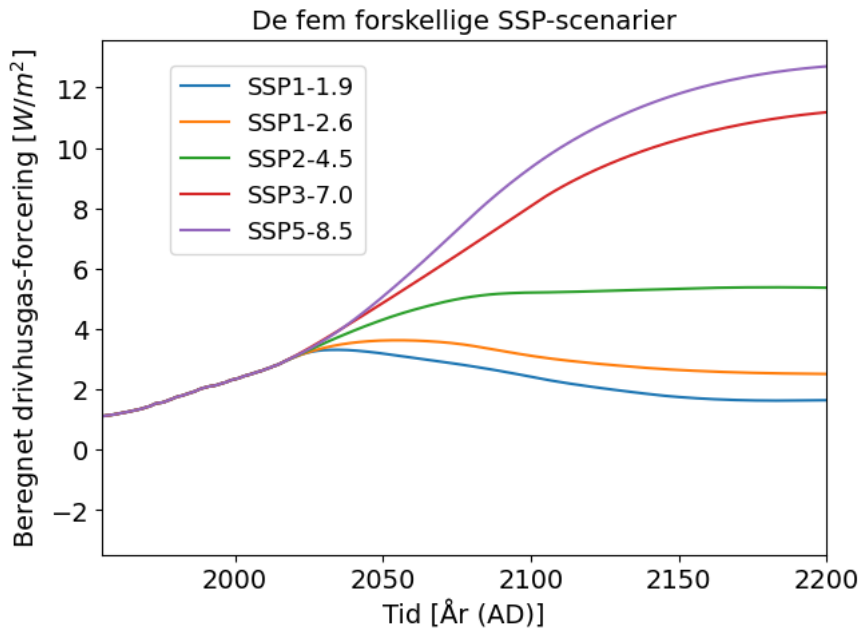
$$\epsilon_0 = 0.624 \pm 0.005 \quad (10)$$

Her antages at $T_0 = (286.8 \pm 0.1) K$ i 1850[8][9] og $A_0 = 0.297 \pm 0.005$, hvortil ophobningsloven bruges til at bestemme en usikkerhed på ϵ_0 . Da epsilon er temperatur afhængig kan et udtryk for epsilon som funktion af temperaturen skrives således, antaget lineær afhængighed

med en hældningskoefficient $k = -0.00411 \pm 0.00005[1]$:

$$\epsilon(T(t)) = \epsilon_0 + k \cdot (T(t) - T_0) \quad (11)$$

Drivhusgasforcingen modelleres på baggrund af data fra de forskellige SSP scenarier, og ligninger der beskriver de individuelle drivhusgassers forceringrater. Figur 5 er opnået ved at indsætte data for de forskellige SSP scenariers drivhusgasudvikling i ligningerne for drivhusgas-forcing [11][7]. Med dette data har vi ved hjælp af interpolate funktionen i python, lavet fem forskellige funktioner til de fem forskellige SSP scenarier, der beskriver drivhusgasforcingen til tiden, $\Delta F(t)$.



Figur 5: Modelleringer af drivhusgas-forcingens udvikling i de forskellige SSP scenarier

Med udtryk for alle parametre i ligning 6, kan temperatur udviklingen fra 1850 til 2200 nu modelleres ved hjælp af Runge-Kutta af 4. orden.

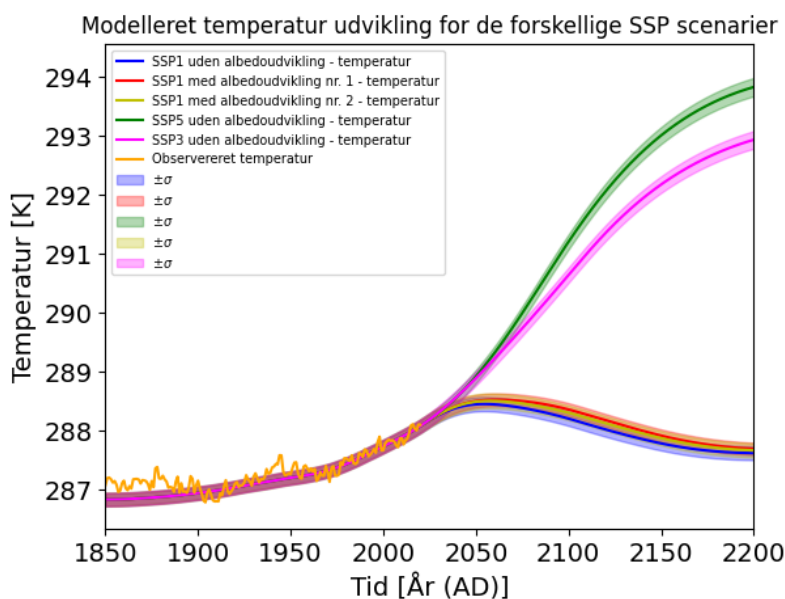
3.2 Monte Carlo simulation

Usikkerheden på modellens beregnede temperatur er svær at fastslå nøjagtigt, da de fleste af usikkerhederne på modellens parametre er ukendte. For at estimere usikkerheden på modellens beregnede temperaturer benyttes en Monte Carlo-simulation, der kan tage højde for de kendte usikkerheder og give et mere realistisk bud på den endelige temperatur. Monte Carlo-simulationen fungerer ved at køre modellen flere gange for at overføre de kendte parametres usikkerheder til den beregnede temperatur. Vi kører modellen 100 gange, hvor

de kendte parametre - havets varmekapacitet, epsilons afhængighed af temperaturen, A_{jord} og temperaturen i år 1850 - ved hver kørsel tilfældigt bestemmes ud fra en normalfordeling med deres bestemte værdi som middelværdi og deres usikkerhed som standardafvigelse (Se evt. appendiks 2). Usikkerheden på ϵ_0 medregnes ikke direkte, da usikkerhederne på ϵ_0 og A_{jord} er afhængige af hinanden. Derfor vælges usikkerheden på A_{jord} som den bestemmende faktor, da den udgør den største usikkerhed. Monte Carlo-simulationen giver os således 100 forskellige bud på havets overfladetemperatur for hver beregnet dato. Disse 100 værdier er også normalfordelte omkring en gennemsnitstemperatur, hvis usikkerhed kan beregnes fra standardafvigelsen mellem de 100 punkter.

4 Resultater

På figur 6, ses de forskellige SSP scenariers temperaturer med usikkerheder, samt SSP1-1.9 sceneriet, hvor albedoen fra solcellerne placeret i ørkenen, er medregnet. Vi ser her at temperaturerne er de samme indtil år 2015, dette giver god mening, da de alle er baseret på samme forcering og albedo indtil år 2015. Herefter ser vi at for SSP5-8.5, stiger temperaturen mest, og her vil der være en temperaturstigning på cirka $6K$. Ved scenarie SSP1-1.9, stiger temperaturen mindst og her vil temperaturen stige med cirka $0.7K$, se tabel 1. På figur 6 ses det at i de to scenarier, hvor vi ikke begrænser for forbrug af CO_2 , stiger temperaturen markant mere end i de tre scenarier, hvor vi stopper med at udlede CO_2 .



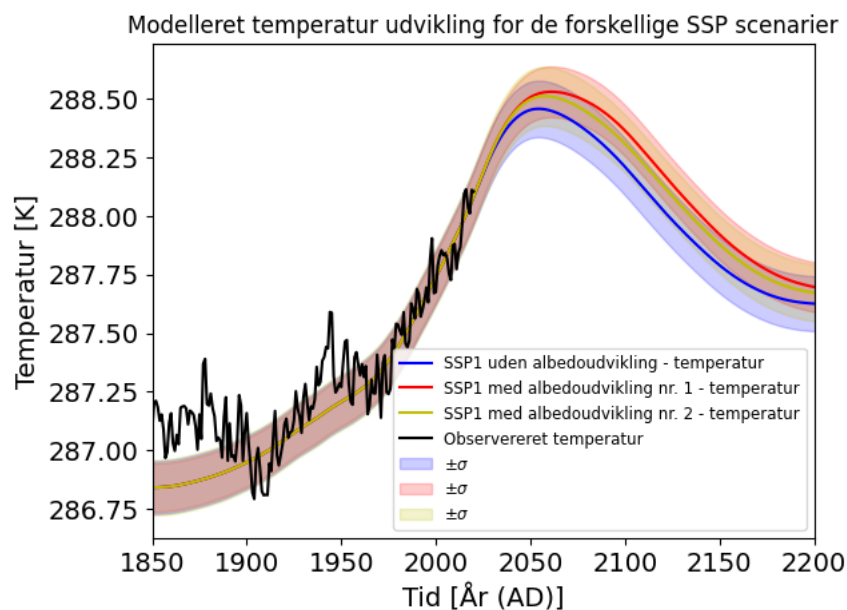
Figur 6: Figuren viser ændringen i temperatur, for de forskellige SSP scenarier, samt de to scenarier hvor albedoen er medregnet.

På figur 7 vises solcellers indvirkning på Jordens albedo tydeligere. Her ses det at SSP1-1.9 uden albedoudvikling, har den laveste makstemperaturændring af scenarierne. SSP1-1.9 med albedoudvikling nr 1, har den højeste makstemperaturændring og SSP1-1.9 med albedoudvikling nr 2, har en makstemperaturændring, der ligger mellem de to. Dette er forventet, da albedoudvikling nr. 1 er modeleret på baggrund af scenariet med den højeste CO_2 udledning (SSP5-8.5), og albedoudvikling nr. 2 er bygget på den næsthøjeste CO_2 udledning (SSP3-7.0). Ved brug af vores model, har vi fundet forskellen mellem makstemperaturændringerne på de tre grafer. Forskellen mellem SSP1-1.9 uden albedo- og SSP1 med albedoudvikling nr. 1 er $0.07K$ og forskellen mellem SSP1-1.9 uden albedo- og SSP1-1.9 med albedoudvikling nr. 2 er $0.05K$.

	SSP1-1.9	SSP1-1.9 A1	SSP1-1.9 A2	SSP3-7.0	SSP5-8.5
Makstemperatur [K]	288.46	288.53	288.51	293.2	294.0
ΔT siden 2015 [K]	0.68	0.75	0.74	5.39	6.26
År	2054	2061	2057	2249	2247
σ [K]	0.1	0.1	0.1	0.2	0.2
Makssolcelleareal [%]	0	0.2898	0.1939	0	0

Tabel 1: Tabel med beskrivende data for de forskellige relevante SSP-scenarier, hvor % angiver solcerllernes areal i procentdel af Jordens areal. Et ekstra decimal medtages i makstemperaturen på trods af usikkerhederne, for at illustrere solcellernes påvirkning på temperaturstigningerne.

I tabellen ovenfor ses makstemperaturerne for de forskellige scenarier, deres usikkerheder, deres maksimale temperaturændring fra 2015, samt hvilket årstal maksimums temperaturen opnås. Yderligere er den maksimale procentdel af Jorden der skal dækkes af solceller i hvert scenarie også beskrevet. Her er A1 albedoudvikling nr. 1 og A2 albedoudvikling nr. 2. For de relevante SSP scenarier kan den totale procentdel af Jordens overfladeareal der kommer til at blive dækket med solceller omregnes til km^2 . Det giver for SSP1-1.9 A1, et areal på $1.478 \cdot 10^6 km^2$, hvilket sammenlignet med Saharas areal på $9.2 \cdot 10^6 km^2$ kun er 16 %. Det samme kan gøres for SSP1-1.9 A2 scenariet, hvor 11 % af Sahara skal dækkes med solceller.



Figur 7: Figur over ændringen i temperatur med usikkerheder, for de forskellige SSP scenarier samt de to scenarier, hvor albedoudviklingen er medregnet.

5 Diskussion

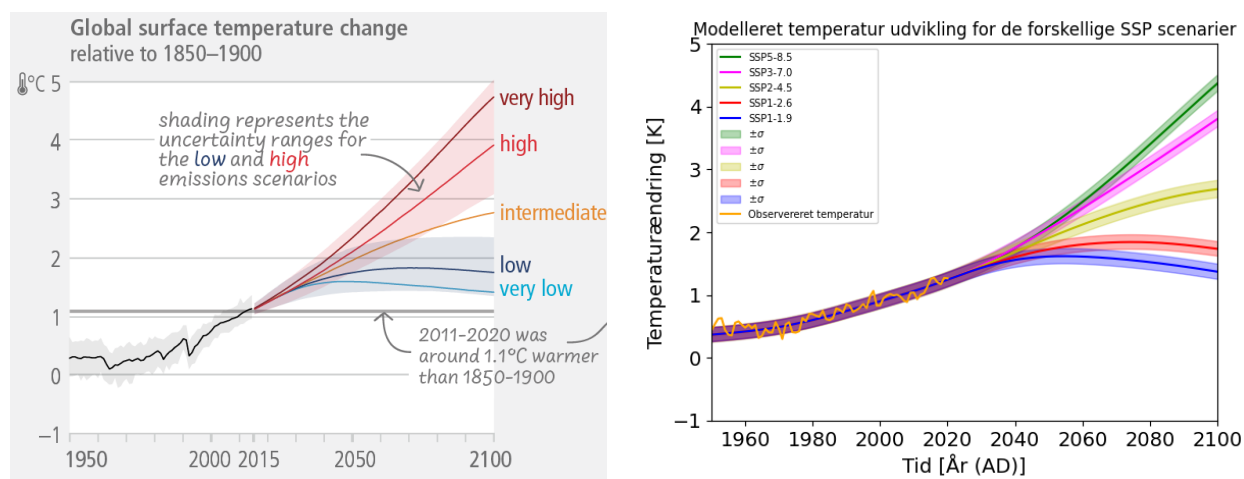
Vores udarbejdede model, bruger mange simplificeringer og der er derfor flere usikkerheder. Målet med denne model er at vise hvilken effekt det vil have på temperaturen, samt albedoen, at erstatte al den fremtidige energi vi får ved brug af fossile og industrielle energier, med solcelleenergi, ved brug af forholdsvis få parameter. Som det ses i vores resultater, vil temperaturen for SSP5-8.5 og SSP3-7.0, ligge betydeligt højere, end de to scenarier, hvor vi erstatter alt CO_2 emissionen, med solceller. Der tages dog ikke højde for temperaturændringernes effekt på Jordens klima, for eksempel vil smelting af Jordens is medføre en lavere albedo, da is har en højere albedo end vand og den mørke landjord der befinder sig under.

I vores model har vi kun usikkerheder på havets varmekapacitet, epsilons afhængighed af temperaturen, A_{jord} og temperaturen i år 1850. Vi har derfor ikke alle usikkerheder medregnet, hvilket resulterer i at modellen er mindre præcis og derved også mindre troværdig, i dens estimat af usikkerheden på temperaturudviklingen. For at vurdere om den bestemte maksimumstemperatur fundet for oceanernes øvre lag er rimelig, kan en analytisk udregning af maksimumstemperaturen til samme tidspunkt udføres. Ved antagelse af at kloden er i energibalance i de givne år, kan ligning 1 og ligning 5 sættes lig med hinanden. Ved brug af ligningens parametre til årstallet 2061, og 2057, kan en maksimumstemperatur på $288.54K$

bestemmes ud fra SSP1-1.9 med albedoudvikling nr. 1 og $288.52K$ ud fra albedoudvikling nr. 2. Da begge værdier falder inden for modellens estimer, taget deres usikkerheder i betragtning, kan modellen alligevel dømmes relativt præcis. Den analytiske metodes meget marginale overskud skyldes at modellen aldrig når energibalance ved disse parametre, da oceantemperaturen har en langsom responsetid på ændringen af energibalancen. Dette medfører at modellen ser en mindre temperaturudvikling, da drivhusgas-forcingen når at falde, før oceanet når at nå dets maksimale temperatur. For at optimere vores model, ville man skulle indføre flere parametre, da der er langt flere faktorer, der er afgørende for om temperaturen stiger, end hvad vi har inkluderet i modellen.

Som det fremgår i vores resultater, vil vi skulle dække 16 % af Sahara i SSP5-8.5 scenariet. Dette er en forholdsvis stor andel af Sahara og er svært at implementere, da det kan have økonomiske og samfundskritiske problemer. Infrastrukturen i Saharas ørken er ikke ligeså god, som for eksempel i Danmark, det vil derfor være svært at bringe strømmen ud til hele verden derfra. Samtidig vil vedligeholdelse af solcellerne, samt forsikring mod tyveri og hærværk være en stor opgave. Finansieringen af solcellerne, på denne skala, vil også blive yderst udfordrende, og højst sandsynligt kræve internationalt samarbejde. Hvis vi kigger på scenarie SSP3-7.0, ser vi at andelen af Sahara der skal dækkes her er mindre, men stadig 11 %, hvilket vil medføre de samme problemer som redegjort for herover.

I IPCC's seneste rapport, er der lavet en opsamlet model der viser hvordan temperaturen vil ændre sig ud fra de fem forskellige SSP scenarier, med usikkerheder på SSP1-2.6 og SSP3-7, se figur 8. Her ses deres modelering over temperaturudviklingen, indtil år 2100. Det



Figur 8: Temperaturudvikling for de fem SSP scenarier, indtil år 2100[5]

ses ved sammenligning af den opsamlede model IPCC bruger, med vores udarbejdede model, at de ligner hinanden yderst meget. Vores model er dog en forsimplet version, i forhold til IPCC's opsamlede model, da vi som nævnt ovenfor, laver flere antagelser samtidig med vi ikke tager højde for alle usikkerheder, for at forsimple beregningerne. En vigtig parameter vores model ikke tager højde for, er svovludviklingen, da svovl er en af de eneste drivhusgasser der reflekterer solens kortbølgede stråling ud mod universet igen. Svovl medvirker derfor til en lavere drivhusgas-forcing, der ultimativt ville have resulteret i en lavere temperaturudvikling, end den vi har i vores model. Det viser sig dog, at oceanets retarderende effekt på temperaturudviklingen, modvirker den forhøjede temperatur, manglen af svovludviklingen medfører. Dette er højst sandsynligt hvorfor vi stadig ser at vores model passer godt til IPCC's opsamlede model.

6 Konklusion

I denne rapport har vi undersøgt, hvordan fremtidens energiforbrug vil påvirke Jordens klima, og hvordan solceller kan bidrage til en mere bæredygtig fremtid. Vi har set, at temperaturen på Jorden vil stige markant i de scenarier, hvor der ikke tages hensyn til begrænsning af CO_2 udledningen, op mod $6K$. Derimod vil temperaturstigningen være meget lavere i de scenarier, hvor udledningen begrænses, faktisk helt nede på $0.7K$ i forhold til 2015.

Vi har undersøgt, hvordan solceller kan påvirke Jordens albedo, og dermed temperaturen. Modellen viser, at solceller i ørkenerne kan medvirke til en lavere temperaturstigning. Og dertil også beregnet, hvor stor en del af Jordens overflade, der skal dækkes med solceller for at kunne dække energibehovet i visse scenarier, henholdsvis 16% og 11%. Vores resultater viser, at solceller kan spille en vigtig rolle i en mere bæredygtig fremtid, men det kræver også, at der tages hensyn til de potentielle udfordringer, som eksempelvis ændringer i Jordens albedo der ultimativt kan forårsage en temperaturændring på $0.07K$. På trods af denne temperaturændring, kan det dog konkluderes, at solceller ikke ligefrem bliver en stor kilde til global opvarmning.

7 Acknowledgements

Dette projekt ville ikke kunne have lade sig gøre uden hjælp fra Bo Møllesø Vinther. Vi vil gerne sige stor tak til Bo, for den gode vejledning, og den store interesse for vores projekt.

Han har hjulpet os godt på vej med materiale, data og modeller, til at lave vores projekt. Det har været en fornøjelse at samarbejde med Bo.

Litteratur

- [1] J. H. Appel, J. S. Kjær, and K. Valeckas. Simulation af temperaturændringer de sidste 800.000 år med en simpel nuldimensionel energibalancemodel. *Bachelorstudiet i fysik Førsteårsprojekt 2021*, (2021-19):15, 2021. Indsendt som en pdf-fil den 8. april 2021.
- [2] R. J. Barlow. *Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences (Manchester Physics Series)*. WileyBlackwell, reprint edition, 1989.
- [3] Department for Energy Security and Net Zero and Department for Business, Energy & Industrial Strategy. Greenhouse gas reporting: conversion factors 2018. Online, June 2018. Last accessed on 27 March 2023.
- [4] P. Goode, J. Qiu, V. Yurchyshyn, J. Hickey, M.-C. Chu, E. Kolbe, C. Brown, and S. Koonin. Earthshine observations of the earth’s reflectance. *Geophysical Research Letters - GEOPHYS RES LETT*, 28:1671–1674, 05 2001.
- [5] D. D. I. G. K. F. A. M. I. . Z. Z. L. Hoesung Lee (Chair), Katherine Calvin (USA). Synthesis report of the ipcc sixth assessment report (ar6). *Intergovernmental Panel on Climate Change, 2023*, 85:1–85, 03 2023.
- [6] Japan Meteorological Agency. Global ocean heat content. <https://www.jma.go.jp/jma/en/Activities/oceanheat.html>, 2023. Accessed: March 27, 2023.
- [7] o. B. V. Jonathan Melcher, Sune Halkjær. Modellering af fremtidens klima med en energibalance-model. 2023.
- [8] P. Jones, M. New, D. Parker, S. Martin, and I. Rigor. Surface air temperature and its changes over the past 150 years. *Reviews of Geophysics*, 37(2):173–199, 5 1999.
- [9] J. Kennedy, N. Rayner, R. Smith, D. Parker, and M. Saunby. Reassessing biases and other uncertainties in sea-surface temperature observations measured in situ since 1850: 1. measurement and sampling uncertainties. *Journal of Geophysical Research: Atmospheres*, 116(D14), 2011.

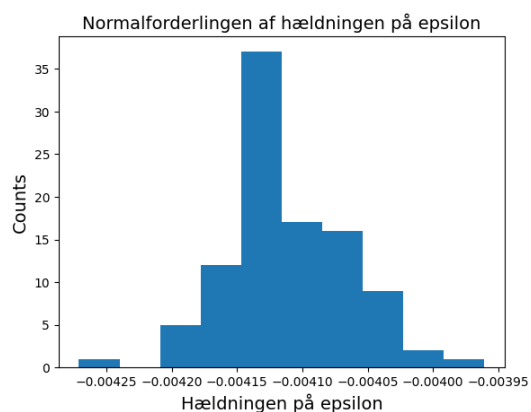
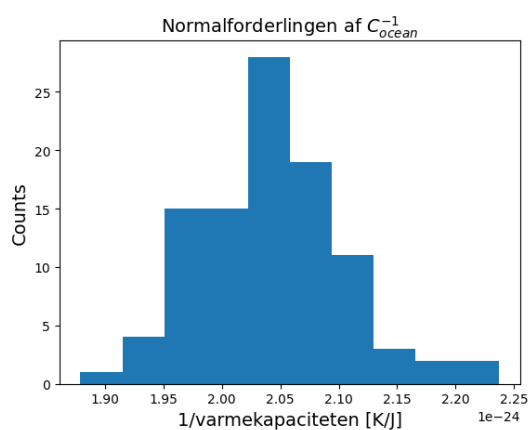
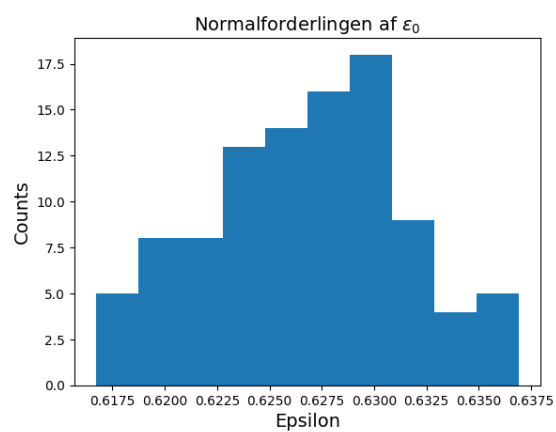
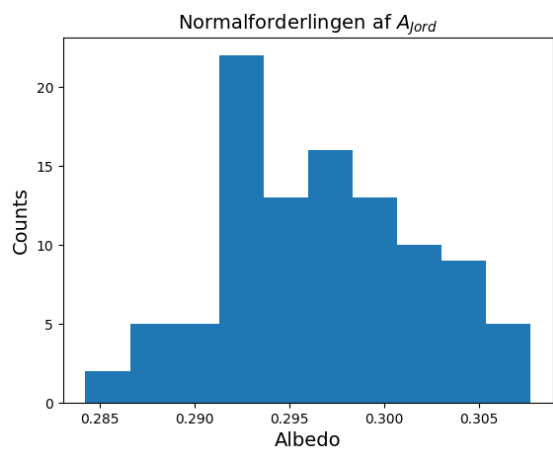
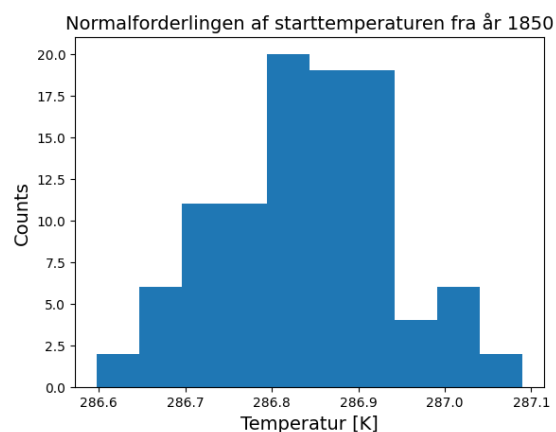
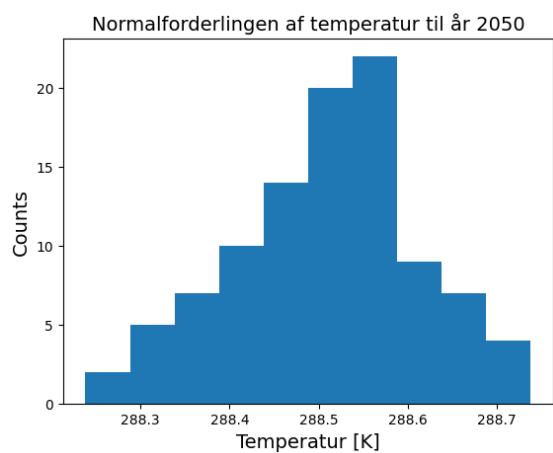
- [10] Y. Li, E. Kalnay, S. Motesharrei, J. Rivas, F. Kucharski, D. Kirk-Davidoff, E. Bach, and N. Zeng. Climate model shows large-scale wind and solar farms in the sahara increase rain and vegetation. *Science*, 361:1019–1022, 09 2018.
- [11] M. Meinshausen, Z. R. J. Nicholls, J. Lewis, M. J. Gidden, E. Vogel, M. Freund, U. Beyerle, C. Gessner, A. Nauels, N. Bauer, J. G. Canadell, J. S. Daniel, A. John, P. B. Krummel, G. Luderer, N. Meinshausen, S. A. Montzka, P. J. Rayner, S. Reimann, S. J. Smith, M. van den Berg, G. J. M. Velders, M. K. Vollmer, and R. H. J. Wang. The shared socio-economic pathway (ssp) greenhouse gas concentrations and their extensions to 2500. *Geoscientific Model Development*, 13(8):3571–3605, 2020.
- [12] J. Stoer and R. Bulirsch. *17.1 Runge-Kutta Method*. Springer, New York, 3rd edition, 2002.

8 Appendiks

8.1 Appendiks 1 - ligninger til beregning af drivhusgas-forcering

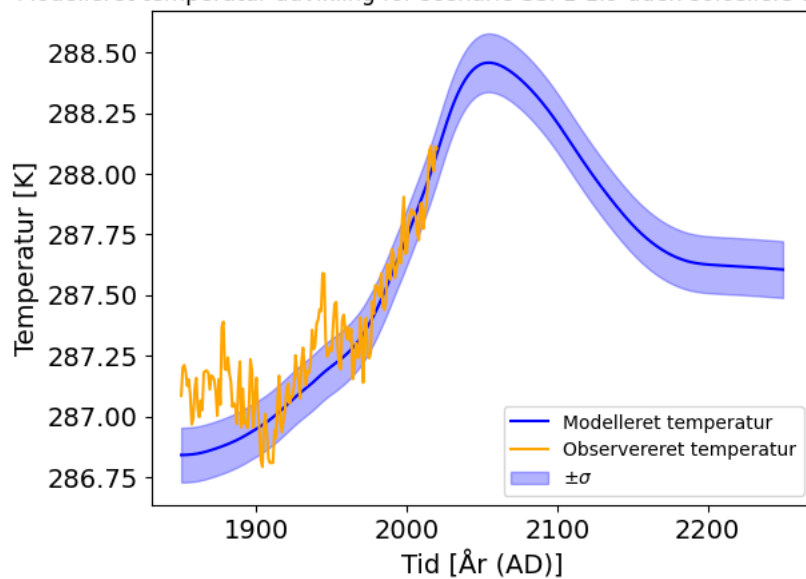
Gas	Simplified Expression	Coefficients error % (W m^{-2})	Maximal absolute fit
CO ₂	$C_{\alpha_{\max}} = C_0 - \frac{b_1}{2a_1} \approx 1808 \text{ ppm}$ $\alpha' = d_1 - \frac{b_1^2}{4a_1}, \text{ for } C > C_{\alpha_{\max}}$ $\alpha' = d_1 + a_1(C - C_0)^2 + b_1(C - C_0), \text{ for } C_0 < C < C_{\alpha_{\max}}$ $\alpha' = d_1, \text{ for } C \leq C_0$ $\alpha_{\text{N}_2\text{O}} = c_1 \cdot \sqrt{N}$ $\text{RF}_{\text{CO}_2} = (\alpha' + \alpha_{\text{N}_2\text{O}}) \cdot \ln\left(\frac{C}{C_0}\right)$	$a_1 = -2.4785 \times 10^{-7} \text{ W m}^{-2} \text{ ppm}^{-2}$ $b_1 = 0.00075906 \text{ W m}^{-2} \text{ ppm}^{-1}$ $c_1 = -0.0021492 \text{ W m}^{-2} \text{ ppb}^{-0.5}$ $d_1 = 5.2488 \text{ W m}^{-2}$ $C_0 = 277.15 \text{ ppm}$	0.11 % (0.0037 W m^{-2})
N ₂ O	$\text{RF}_{\text{N}_2\text{O}} = (a_2\sqrt{C} + b_2\sqrt{N} + c_2\sqrt{M} + d_2) \cdot (\sqrt{N} - \sqrt{N_0})$	$a_2 = -0.00034197 \text{ W m}^{-2} \text{ ppm}^{-1}$ $b_2 = 0.00025455 \text{ W m}^{-2} \text{ ppb}^{-1}$ $c_2 = -0.00024357 \text{ W m}^{-2} \text{ ppb}^{-1}$ $d_2 = 0.12173 \text{ W m}^{-2} \text{ ppb}^{-0.5}$ $N_0 = 273.87 \text{ ppb}$	1.5 % (0.0059 W m^{-2})
CH ₄	$\text{RF}_{\text{CH}_4} = (a_3\sqrt{M} + b_3\sqrt{N} + d_3) \cdot (\sqrt{M} - \sqrt{M_0})$	$a_3 = -8.9603 \times 10^{-5} \text{ W m}^{-2} \text{ ppb}^{-1}$ $b_3 = -0.00012462 \text{ W m}^{-2} \text{ ppb}^{-1}$ $d_3 = 0.045194 \text{ W m}^{-2} \text{ ppb}^{-0.5}$ $M_0 = 731.41 \text{ ppb}$	0.55 % (0.0032 W m^{-2})

8.2 Appendiks 2 - Normalfordeling af parametre med usikkerheder



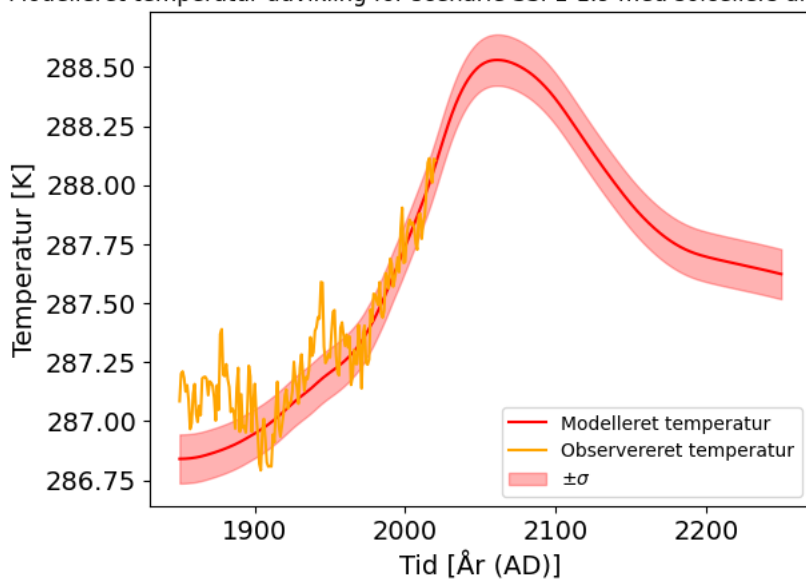
8.3 Appendiks 3 - Pythonkodens figurer

Modelleret temperatur udvikling for scenarie SSP1-1.9 uden solcellers albedo effect



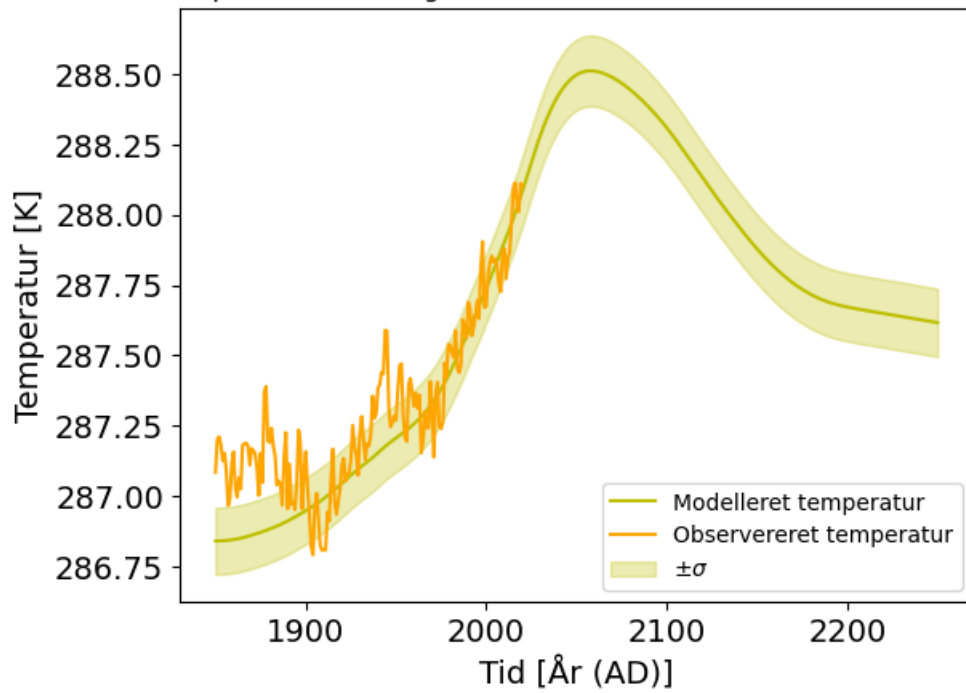
Figur 12

Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 1



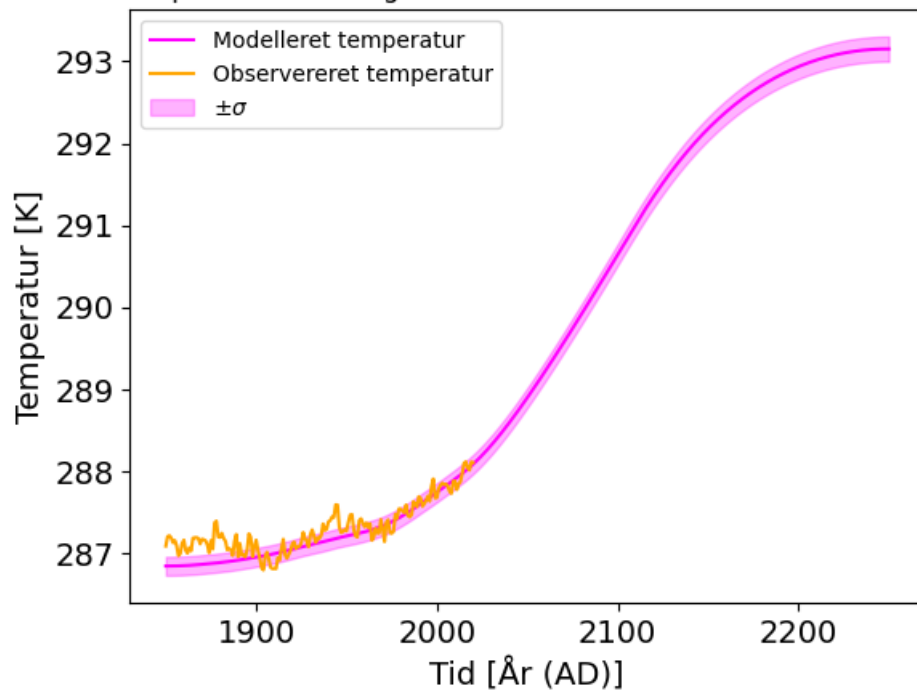
Figur 13

Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 2



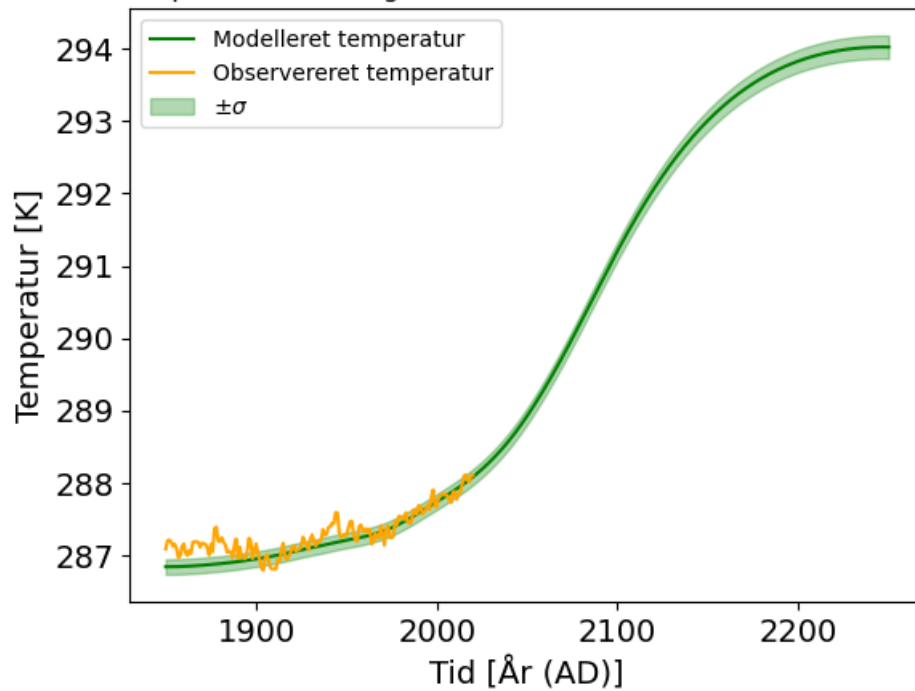
Figur 14

Modelleret temperatur udvikling for scenarie SSP3-7.0 uden solcellers albedo effect



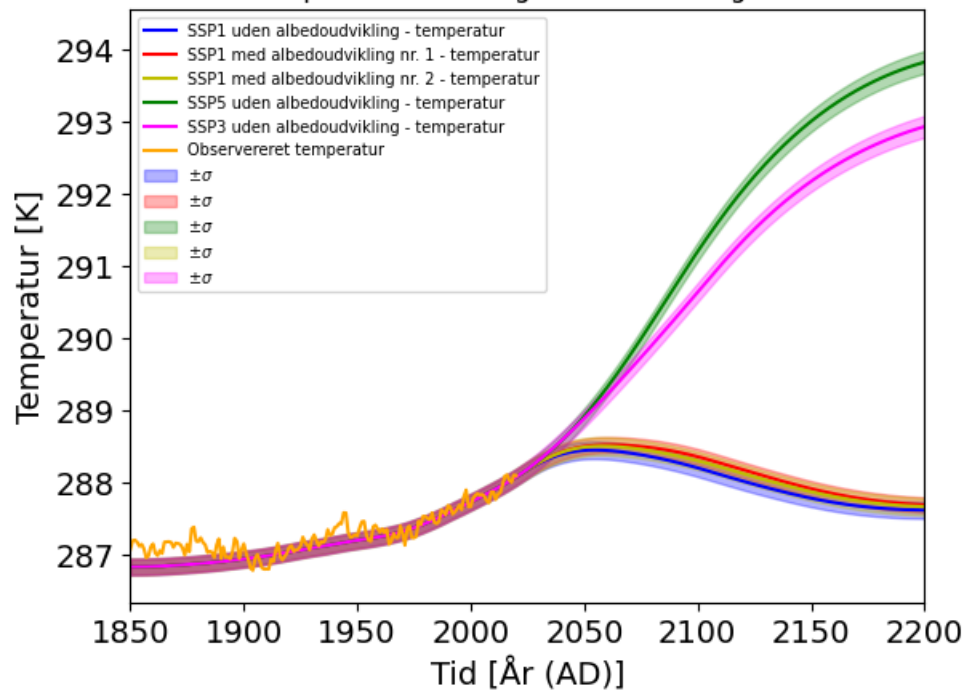
Figur 15

Modelleret temperatur udvikling for scenarie SSP5-8.5 uden solcellers albedo effect

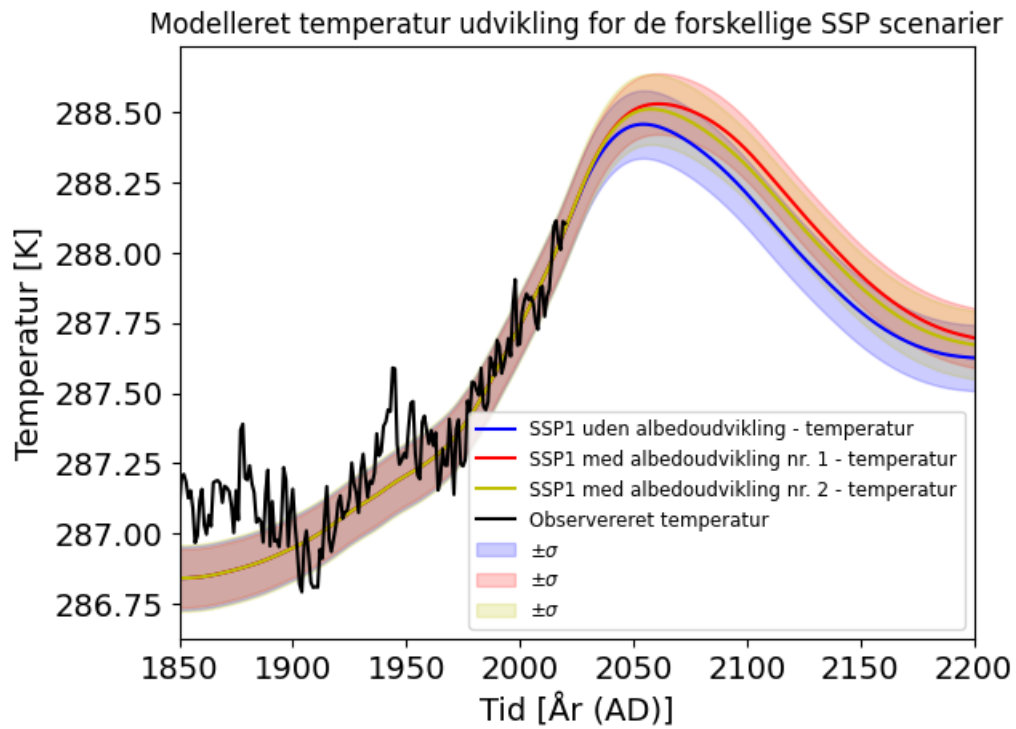


Figur 16

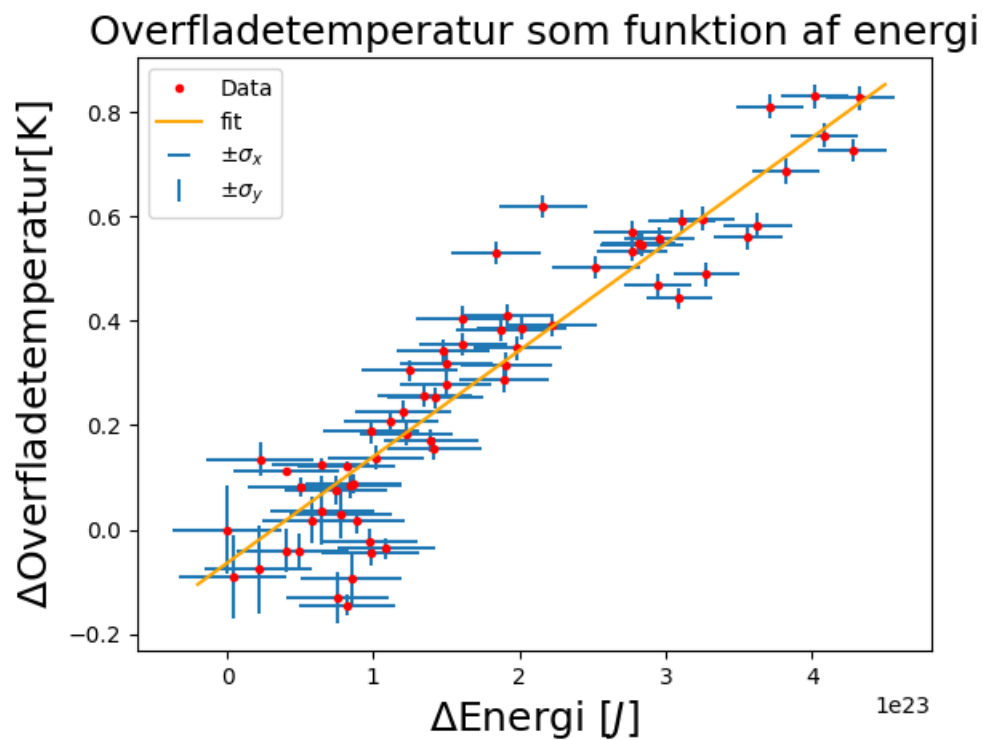
Modelleret temperatur udvikling for de forskellige SSP scenarier



Figur 17

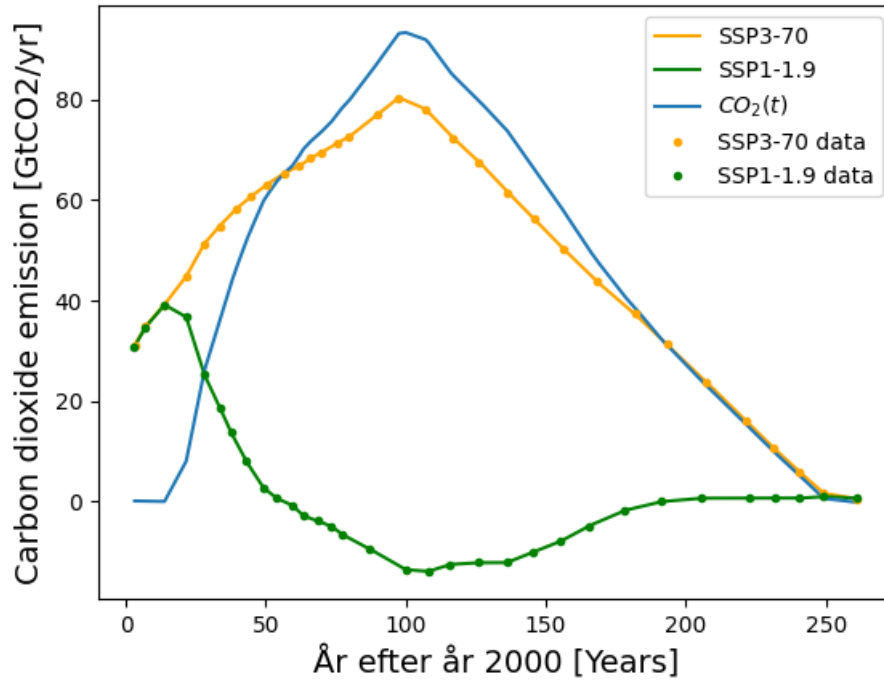


Figur 18



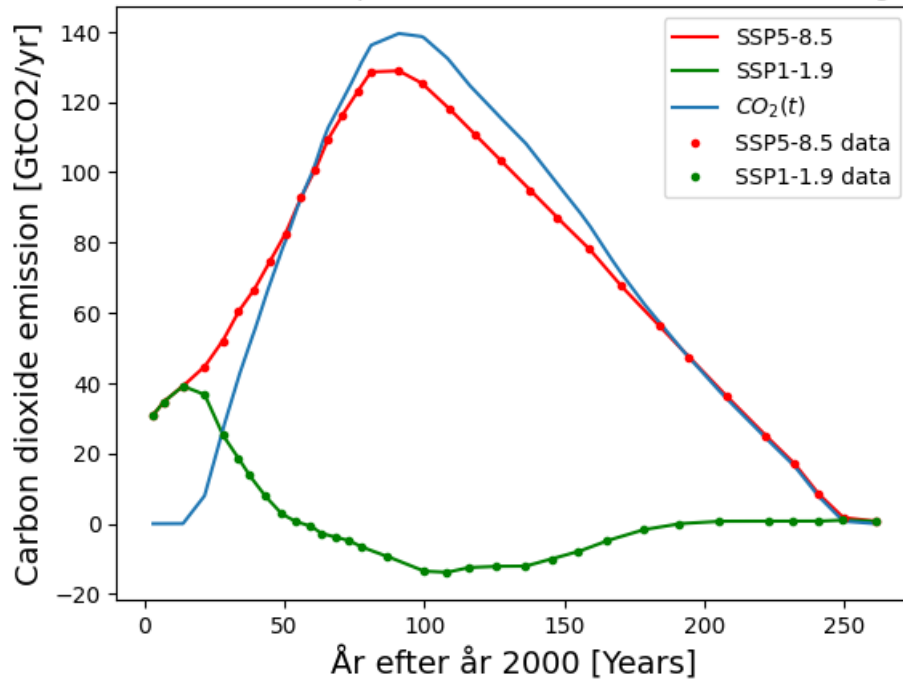
Figur 19

Model over forskellen på CO_2 emission i SSP scenario 3-70 og 1-1.9

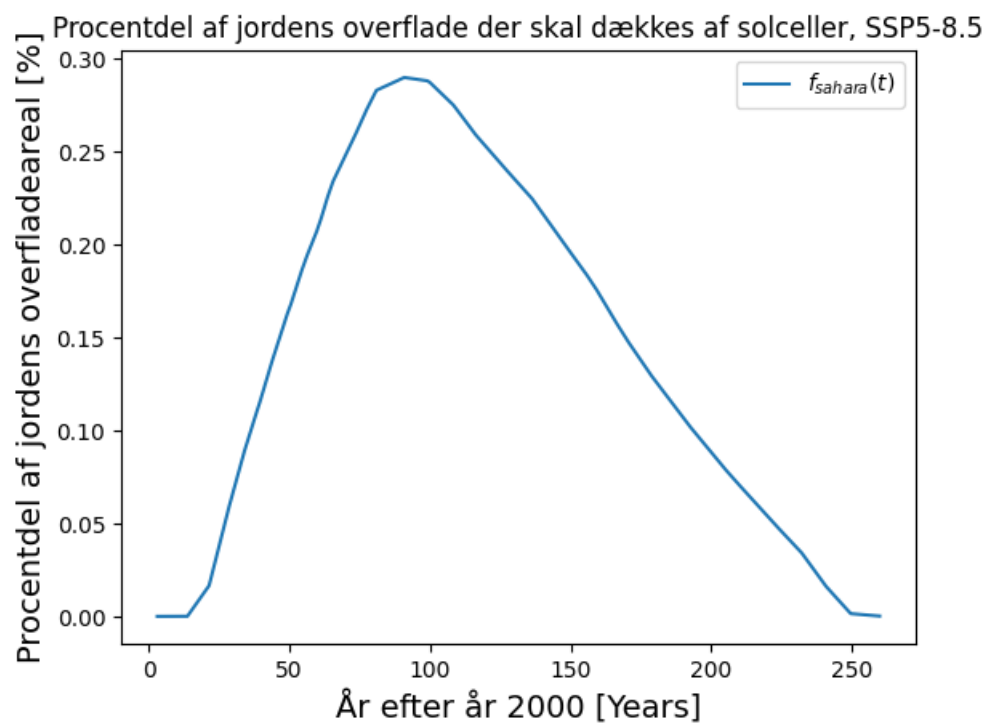


Figur 20

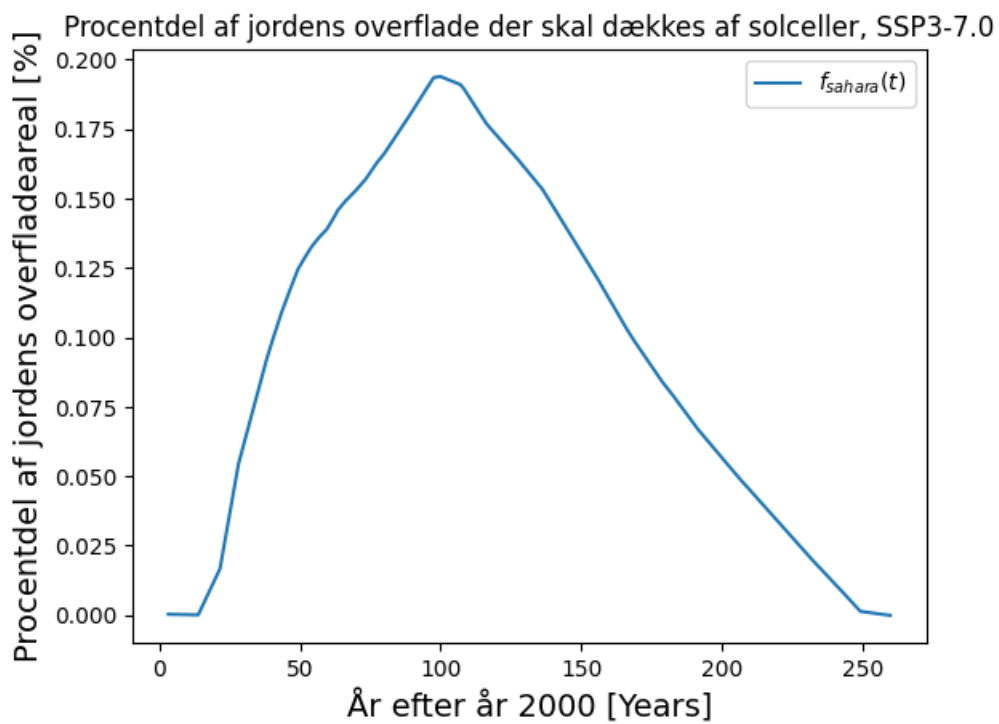
Model over forskellen på CO_2 emission i SSP scenario 5-8.5 og 1-1.9



Figur 21



Figur 22



Figur 23

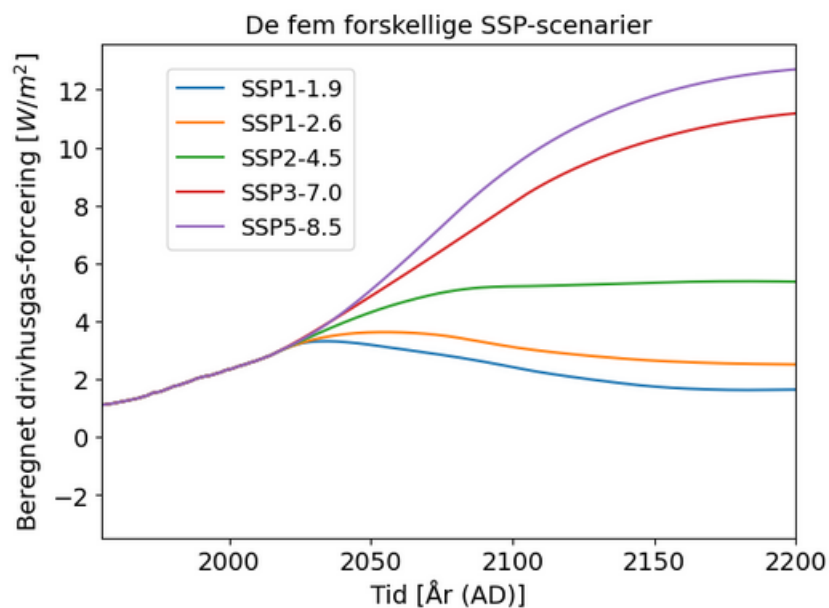
8.4 Appendiks 4 - Rå python kode

```
In [1]: # Indlæs biblioteker
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d
from scipy.optimize import curve_fit

In [25]: # Indlæs SSP drivhusgas-forceringsscenarier og plotter dem.
# Disse scenarier anvendes til IPCC's seneste rapport og bruges til at definere menneskeskabt strålingsforcering.

SSP=np.loadtxt('SSP_Paleo_GHG_Forcing_Scenarios.txt') # Indlæs SSP scenarier
CO2DATA = np.genfromtxt('co2udvikling.csv',skip_header=1, delimiter=',')
OceanHeat = np.genfromtxt('ohc_global_1955.txt',skip_header=8, skip_footer=2)
SeaTemp = np.genfromtxt('sea-surface-temperature-anomaly.csv',skip_header=105, skip_footer=514, delimiter=',')

fig_Profile , ax_Profile = plt.subplots ( figsize =(7 , 5) ) # Åbner en figur
l1, =ax_Profile.plot(SSP[:,0],SSP[:,1]) # Tegner Scenario SSP1-1.9
l2, =ax_Profile.plot(SSP[:,0],SSP[:,2]) # Tegner Scenario SSP1-2.6
l3, =ax_Profile.plot(SSP[:,0],SSP[:,3]) # Tegner Scenario SSP2-4.5
l4, =ax_Profile.plot(SSP[:,0],SSP[:,4]) # Tegner Scenario SSP3-7.0
l5, =ax_Profile.plot(SSP[:,0],SSP[:,5]) # Tegner Scenario SSP5-8.5
plt.xlim(2000,2200)
fig_Profile.legend((l1,l2,l3,l4,l5),('SSP1-1.9','SSP1-2.6','SSP2-4.5','SSP3-7.0','SSP5-8.5'),loc=[0.2,0.6],fontsize=13)
ax_Profile.set_xlabel('Tid [År (AD)]',fontsize=14)
ax_Profile.set_ylabel('Beregnet drivhusgas-forcering '+r'[$W/m^2$]',fontsize=14)
ax_Profile.tick_params(which='major',labelsize=14)
ax_Profile.set (xlim=(1955,2200))
ax_Profile.set_title("De fem forskellige SSP-scenarier", fontsize=14)
plt.savefig('SSP_GHG_Forcing_Scenarios.png',dpi=300, bbox_inches='tight')
```



Figur 24

```

In [54]: f = interp1d(CO2DATA[:,0],CO2DATA[:,1]) # Lineær interpolations-funktion defineres fro det valgte scenarie
f2 = interp1d(CO2DATA[:,2],CO2DATA[:,3])
f3 = interp1d(CO2DATA[:,4],CO2DATA[:,5])

#Funktion der giver Gigaton carbondioxid vores solceller skal erstatte efter år 2000 til år 2250
'''
def f(x):
    return 43.8809108-0.822596154*x+0.0693815243*x**2-0.000385655734*x**3-5.58257207*10**(-6)*x**4+6.56002888*10**(-8)*x**5-2.4243
def f2(x):
    return 35.0322691+2.15186087*x-0.135149413*x**2+0.00251883660*x**3-2.32504086*10**(-5)*x**4+1.16041235*10**(-7)*x**5-2.98410408
def f3(x):
    return 37.0380686+0.834320771*x+0.00368447139*x**2-8.31505501*10**(-5)*x**3+1.09554915*10**(-7)*x**4+1.02899567*10**(-9)*x**5-
'''
ConversionF = 0.21971830985*3.6641412039
def CO2(x):
    return (f(x)-f2(x))*ConversionF
def CO22(x):
    return (f3(x)-f2(x))*ConversionF
def E(x):
    return (CO2(x)*10**12)*4.289

t1 = np.arange(3.1, 260, 0.1)

plt.plot(t1,f(t1)*ConversionF,label="SSP5-8.5",color="r")
plt.plot(t1,f2(t1)*ConversionF,label="SSP1-1.9",color="g")
plt.plot(t1,CO2(t1),label="$CO_2(t)$")
plt.plot(CO2DATA[:,0], CO2DATA[:,1]*ConversionF,'.', label="SSP5-8.5 data",color="r")
plt.plot(CO2DATA[:,2], CO2DATA[:,3]*ConversionF,'.', label="SSP1-1.9 data",color="g")
plt.legend()
plt.title("Model over forskellen på $CO_2$ emission i SSP scenario 5-8.5 og 1-1.9")
plt.xlabel("År efter år 2000 [Years]", fontsize=14)
plt.ylabel("Carbon dioxide emission [GtCO2/yr]", fontsize=14)
plt.show()

plt.plot(t1,f3(t1)*ConversionF,label="SSP3-70",color="orange")
plt.plot(t1,f2(t1)*ConversionF,label="SSP1-1.9",color="g")
plt.plot(t1,CO22(t1),label="$CO_2(t)$")
plt.plot(CO2DATA[:,4], CO2DATA[:,5]*ConversionF,'.', label="SSP3-70 data",color="orange")
plt.plot(CO2DATA[:,2], CO2DATA[:,3]*ConversionF,'.', label="SSP1-1.9 data",color="g")
plt.legend()
plt.title("Model over forskellen på $CO_2$ emission i SSP scenario 3-70 og 1-1.9")
plt.xlabel("År efter år 2000 [Years]", fontsize=14)
plt.ylabel("Carbon dioxide emission [GtCO2/yr]", fontsize=14)
plt.show()

FS=1368 # Solindstråling i W/m**2
StS = 8765.8 # Hours in a year
SE = 0.135 # Solcellers effektivitet

EPS = (StS*(FS/4)*SE)/1000 # Energi produceret pr. m^2 solceller i sahara om året i kwh

def AS(x): # Berenger hvor mange km^2 der skal dækkes med solceller for at dække energibehov
    return (E(x)/EPS)*10**-6

def f_sahara(x): # Beregner andelen af jorden der skal dækkes med solceller
    return AS(x)/51010000
def f_sahara2(x): # Beregner andelen af jorden der skal dækkes med solceller
    return (((CO22(x)*10**12)*4.289)/EPS)*10**-6)/51010000

```

Figur 25

```

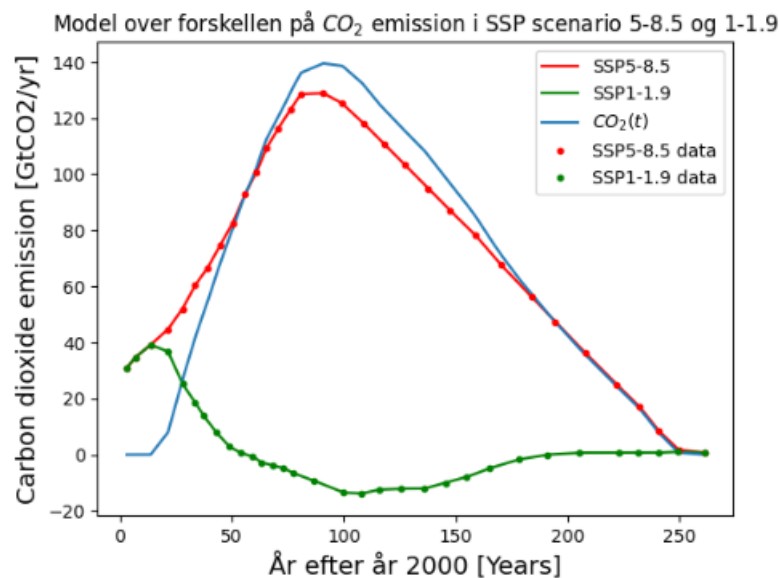
plt.plot(t1,f_sahara(t1)*100,label="$f_{sahara}(t)$")
plt.legend()
plt.title("Procentdel af jordens overflade der skal dækkes af solceller, SSP5-8.5")
plt.xlabel("År efter år 2000 [Years]", fontsize=14)
plt.ylabel("Procentdel af jordens overfladeareal [%]", fontsize=14)
plt.show()

plt.plot(t1,f_sahara2(t1)*100,label="$f_{sahara}(t)$")
plt.legend()
plt.title("Procentdel af jordens overflade der skal dækkes af solceller, SSP3-7.0")
plt.xlabel("År efter år 2000 [Years]", fontsize=14)
plt.ylabel("Procentdel af jordens overfladeareal [%]", fontsize=14)
plt.show()
scenarios = 1 # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios])
print(f_sahara(61), f_sahara2(57))
print(SSPint(2061)-SSPint(1850))
print(SSPint(2057)-SSPint(1850))

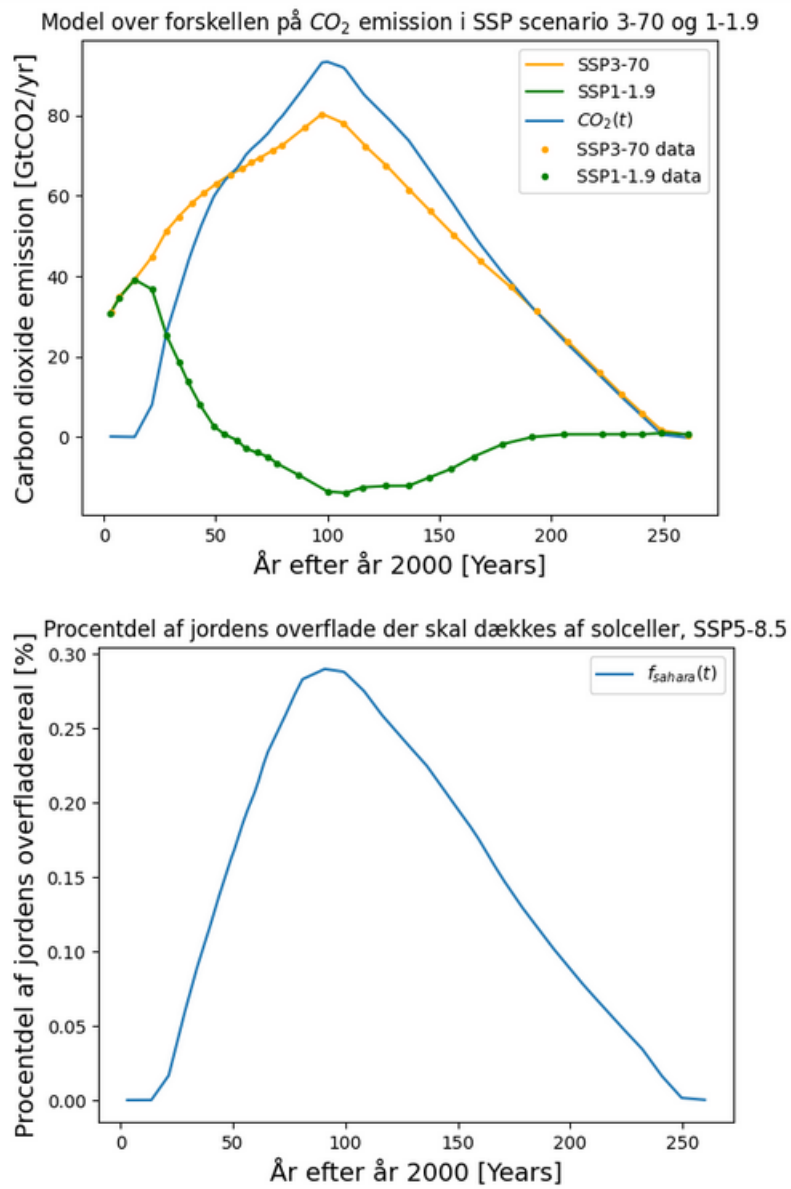
print(f_sahara(14.1)*100)

index = (f_sahara(t1)*100).argmax()
max_valuef = (f_sahara(t1)*100)[index]
year = t1[index]
print(max_valuef, year)
index = (f_sahara2(t1)*100).argmax()
max_valuef2 = (f_sahara2(t1)*100)[index]
year = t1[index]
print(max_valuef2, year)

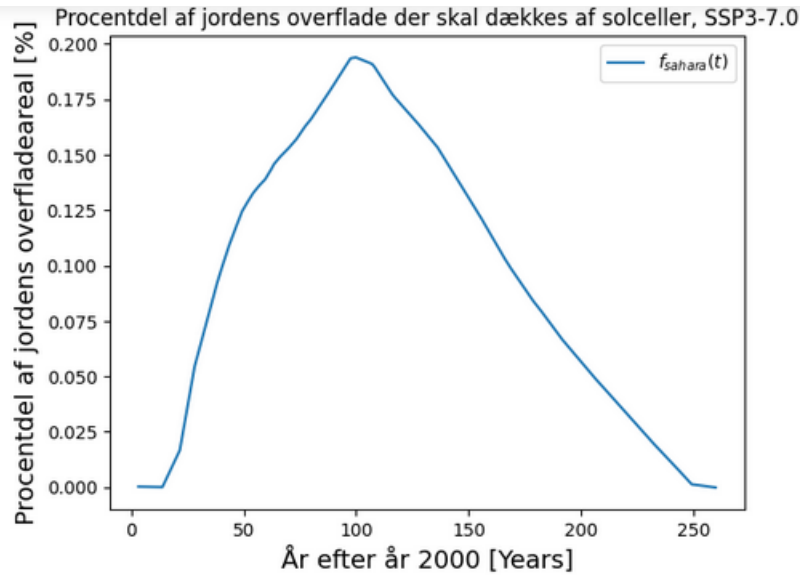
```



Figur 26



Figur 27



```
0.002123377184741762 0.0013620882556025983
2.7688
2.8228
0.0006980953235503217
0.289822079021685 90.80000000000007
0.1938506229140758 99.80000000000008
```

```
In [45]: J = OceanHeat[:,3]*10**22
Te = SeaTemp[:,3]-SeaTemp[0,3]

U_J = np.array(OceanHeat[:,4]*10**22, dtype=np.float64)
U_T = (abs(SeaTemp[:,5])-abs(SeaTemp[:,4]))/2

def linear(x, m, c):
    return m*x + c

#guess = [1000, 1000, 0.94, 1.06, 0.1, 100, 100]
#names = ['A1', 'A2', 'mu1', 'mu2', 'sigma', 'm', 'c']

par, cov = curve_fit(linear, J, Te, sigma=U_T, absolute_sigma=True)
a, d = par
a_err, d_err = np.sqrt(np.diag(cov))

par, cov = curve_fit(linear, J, Te, sigma=np.sqrt(U_T**2+(U_J*a)**2), absolute_sigma=True)
m, c = par
m_err, c_err = np.sqrt(np.diag(cov))

print(cov)
print(np.diag(cov))
X = np.linspace(-0.2*10**23, 4.5*10**23, 1000)
y_model = linear(X, *par)

fig = plt.figure()
```

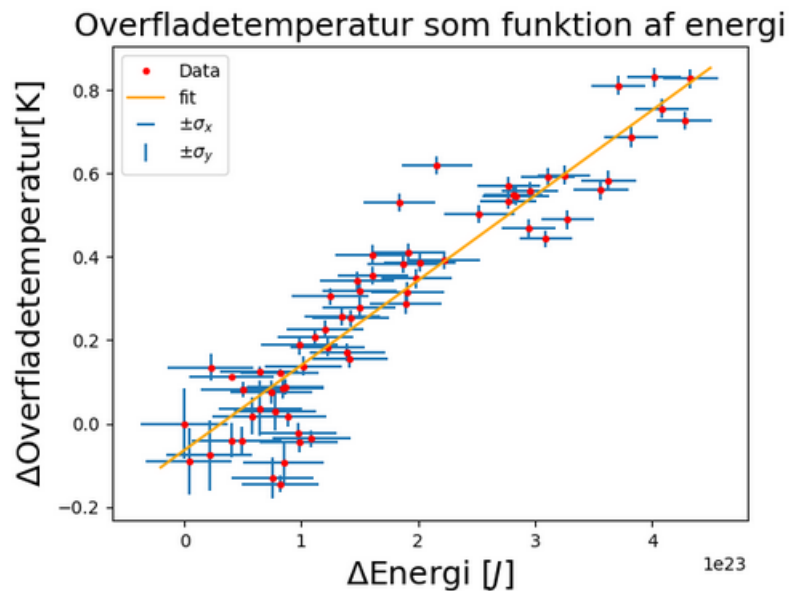
Figur 28

```

fig = plt.figure()
ax = plt.gca()
plt.errorbar(J, Te, xerr=U_J, fmt='none', label = "$\pm \sigma_x$")
plt.errorbar(J, Te, yerr=U_T, fmt='none', label = "$\pm \sigma_y$")
plt.plot(J, Te, '.', label = "Data", color="r")
plt.plot(X, y_model, label = "fit", color = "orange")
plt.legend()
plt.xlabel(r"$\Delta$Energi [J]", fontsize=18)
plt.ylabel(r"$\Delta$Overfladetemperatur[K]", fontsize=18)
plt.title(f"Overfladetemperatur som funktion af energi", fontsize=18)
plt.figure()
print(np.sum(((Te-linear(J,*par))**2)/(U_T**2+(U_J*a)**2))/(len(J)-2))
print(m, m_err)

[[ 3.64063270e-51 -7.77581242e-28]
 [-7.77581242e-28  2.18011134e-04]]
[3.64063270e-51  2.18011134e-04]
2.560901679071095
2.0381928677806174e-24  6.033765570028535e-26

```



<Figure size 640x480 with 0 Axes>

```

In [5]: eps0 = 0.6263779897
eps0_err = 0.0008

deps_dT=-0.00411 # Andring i emissivitet per K temperaturændring
deps_dT_err=0.00005

REarth=6371e3 # Radius af Jordens i m

```

Figur 29

```

dWater=150      # Dybden af det vel blandede lag af havet i m
AOcean=0.05     # Gennemsnitlig albedo af havet
ACloud=0.5      # Gennemsnitlig albedo af skyer
Asolceller=0.1
Aørken=0.4
#(AOcean+ACloud)/2
A0 = 0.297
A0_err = 0.005
def A(x):
    return A0+(Asolceller-Aørken)*f_sahara(x) # Her antages at skyer dækker halvdelen af overfladen og gennemsnitlig albedo udregnes
def A2(x):
    return A0+(Asolceller-Aørken)*f_sahara2(x)
rhoWater=1000   # Densitet af vand i kg/m**3
cWater=4.18e3   # Specifik varmekapacitet af vand i J/(kg)/(K)

COcean1 =dWater*cWater*rhoWater*4*np.pi*REarth**2 # Total varmekapacitet af den velblandede del af verdenshavet
COcean = 1/m
C = m

T0=286.8405     # Start temperatur for Ocean-Jord planeten
T0_err = 0.11511 # Usikkerhed på start temp

SBconst=5.67e-8 # Stefan-Boltzmanns konstant i W/(m**2)/(K**4)

secyear=365.25*24*60*60 # Antal sekunder i et år
t0 = 1850        # Start år

N=4000          # Antal tidsskridt
dt=1/10         # Tidsskridt-størrelse i år
t = np.linspace(t0, t0 + N*dt, N) #Tids-array

```

```

In [6]: #Modellen
def Tidsskridt(V):
    T1, EOcean1, t = V[0], V[1], V[2]
    if eps_temp_dep:
        # temperaturafhængig emissivitet
        eps = eps0 + deps_dT*(T1-T0) #
    else:
        # konstant emissivitet
        eps = eps0

    if human_ghg_on:
        # Indlæsning af drivhusgas-forcering og udregning af, hvor meget den er øget siden tiden t0, hvor modelkørslen startes
        dF = SSPint(t)-SSPint(t0)

    else:
        # Ingen drivhusgas-forcering
        dF = 0

    if A_nr_1_on:
        if solcelle_vækst_on:
            # Udvikling af albedoeffekten grundet solceller
            if t<2014 or t>2260:
                ESW = FS*(1-A0)*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes
            else:
                ESW = FS*(1-A(t-2000))*np.pi*REarth**2*dt*secyear
        else:
            #Ingen udvikling af albedo effekten
            ESW = FS*(1-A0)*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes

```

Figur 30

```

else:
    if solcelle_vækst_on:
        # Udvikling af albedoeffekten grundet solceller
        if t<2014 or t>2260:
            ESW = FS*(1-A0)*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes
        else:
            ESW = FS*(1-A2(t-2000))*np.pi*REarth**2*dt*secyear
    else:
        #Ingen udvikling af albedo effekten
        ESW = FS*(1-A0)*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes

# Beregning af den Langbølge-stråling fra havets overflade, der passerer gennem atmosfæren til tidsskridt "i"
ELW=((eps)*SBconst*(T1**4)-dF) *4*np.pi*(REarth**2)*dt*secyear

NEOcean=ESW-ELW+EOcean1 # Beregning af havets varme indhold for tidsskridt "i"
dEOcean=NEOcean-EOcean1
dT=NEOcean*C-T1 # Beregning af ocean-temperaturen for tidsskridt "i"
return np.array([dT, dEOcean, dt])

#Runge kuta 4. orden fremskrivning
def RK4(V):
    K1 = Tidsskridt(V)
    K2 = Tidsskridt(V+K1/2)
    K3 = Tidsskridt(V+K2/2)
    K4 = Tidsskridt(V+K3)
    K = (K1 + 2*K2 + 2*K3 +K4)/6
    return K

```

```

In [58]: #DEFINITION AF SENARIO: SSP1-1.9 med udvikling af solceller

#ESW = FS*(1-A(15))*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes
# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 1 # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fro det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = True #Tænd og sluk for udvikling af solceller
A_nr_1_on = True #Tænd og sluk for A1 eller A2

#TEMPERATUR BEREGNING
T10 = np.zeros(N)
EOcean10 = np.zeros(N)
C = m
T0=286.8405
eps0 = 0.6263779897
deps_dT=-0.00411
A0 = 0.297
T10[0]=T0 # Initialisering af temperatur
EOcean10[0]=T0/C # Initialisering af havets varme indhold

for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T10[i], EOcean10[i], t[i]])
    dRK4 = RK4(P)
    T10[i+1], EOcean10[i+1]= T10[i]+dRK4[0],EOcean10[i]+dRK4[1]

```

Figur 31

```

#USIKKERHEDS BEREGNING
B = 100
T1 = np.zeros((B,N))
EOcean1 = np.zeros(N)
stT = []
stA = []
stC = []
stEps = []
stEdt = []

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dt=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    stT.append(T0)

    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dt_err = np.random.normal(deps_dt,deps_dt_err)
    stA.append(A0)
    stC.append(C)
    stEps.append(eps0)
    stEdt.append(deps_dt_err)

    T1[j,0]=T0 # Initialisering af temperatur
    EOcean1[0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T1[j,i], EOcean1[i], t[i]])
        dRK4 = RK4(P)
        T1[j,i+1], EOcean1[i+1]= T1[j,i]+dRK4[0],EOcean1[i]+dRK4[1]

U_T1 = np.zeros(N)
T1_up = np.zeros(N)
T1_down = np.zeros(N)

for i in range(N):
    U_T1[i] = np.std(T1[:,i])
    T1_up[i] = T10[i] + U_T1[i]
    T1_down[i] = T10[i] - U_T1[i]

```

```

In [63]: #PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

plt.plot(t,T10, label = "Modelleret temperatur", color = 'r')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T10[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t,T1_down,T1_up, label = "$\pm \sigma$", color = 'r', alpha=0.3)
plt.legend()
plt.title("Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 1")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

```

Figur 32

```

plt.show()
index = T10.argmax()
max_value1 = T10[index]
max_u1 = U_T1[index]
year = t[index]
print(max_value1,max_u1, year)
print(T10[index]-T10[1523])

plt.hist(stT)
plt.xlabel('Temperatur [K]', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af starttemperaturen fra år 1850", fontsize=14)
plt.show()

plt.hist(stA)
plt.xlabel('Albedo', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af $A_{Jord}$", fontsize=14)
plt.show()

plt.hist(stC)
plt.xlabel('1/varmekapaciteten [K/J]', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af $C_{ocean}^{-1}$", fontsize=14)
plt.show()

plt.hist(stEps)
plt.xlabel('Epsilon', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af $\epsilon_0$", fontsize=14)
plt.show()

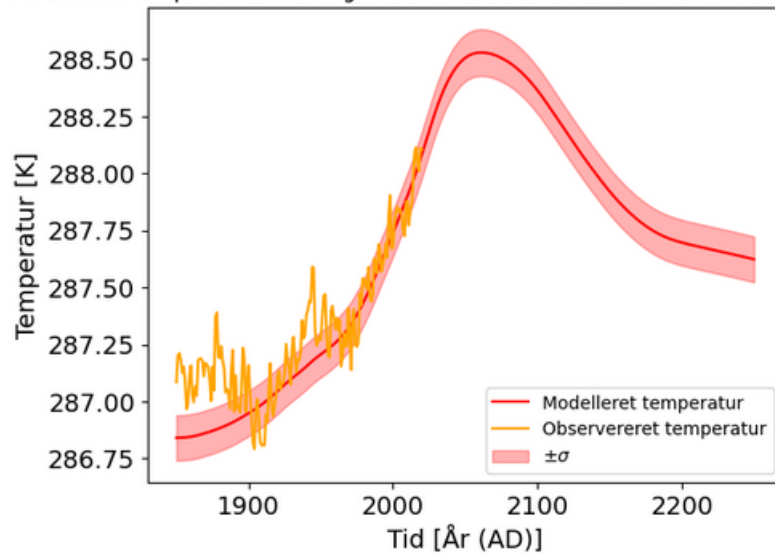
plt.hist(stEdt)
plt.xlabel('Hældningen på epsilon', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af hældningen på epsilon", fontsize=14)
plt.show()

plt.hist(T1[:,2000])
plt.xlabel('Temperatur [K]', fontsize=14)
plt.ylabel('Counts', fontsize=14)
plt.title("Normalfordelingen af temperatur til år 2050", fontsize=14)
plt.show()

```

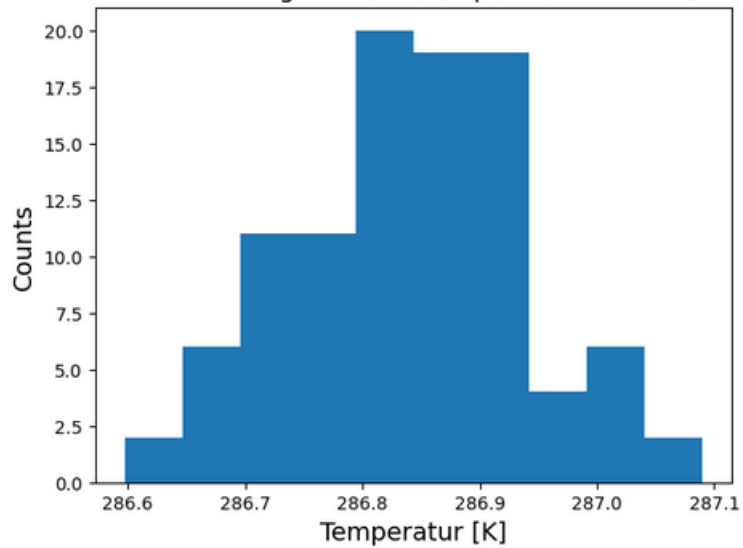
Figur 33

Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 1

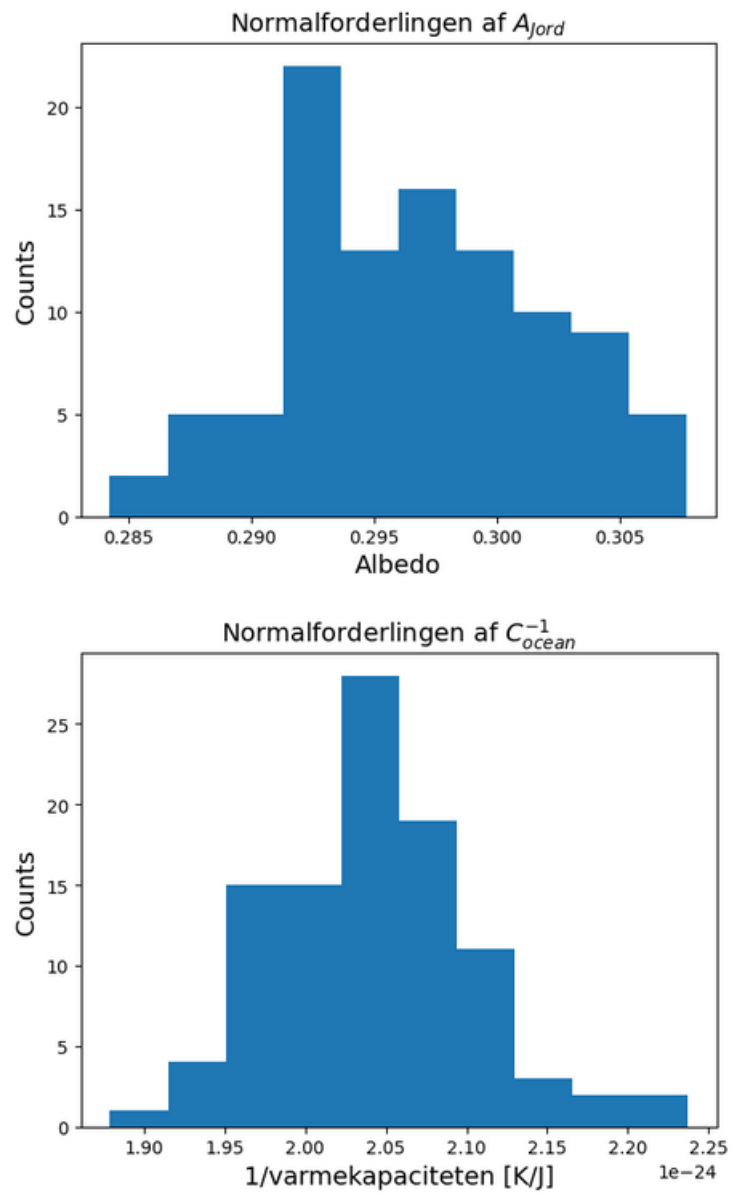


288.5300210392377 0.10238887888004418 2061.0527631907976
0.7543235025411832

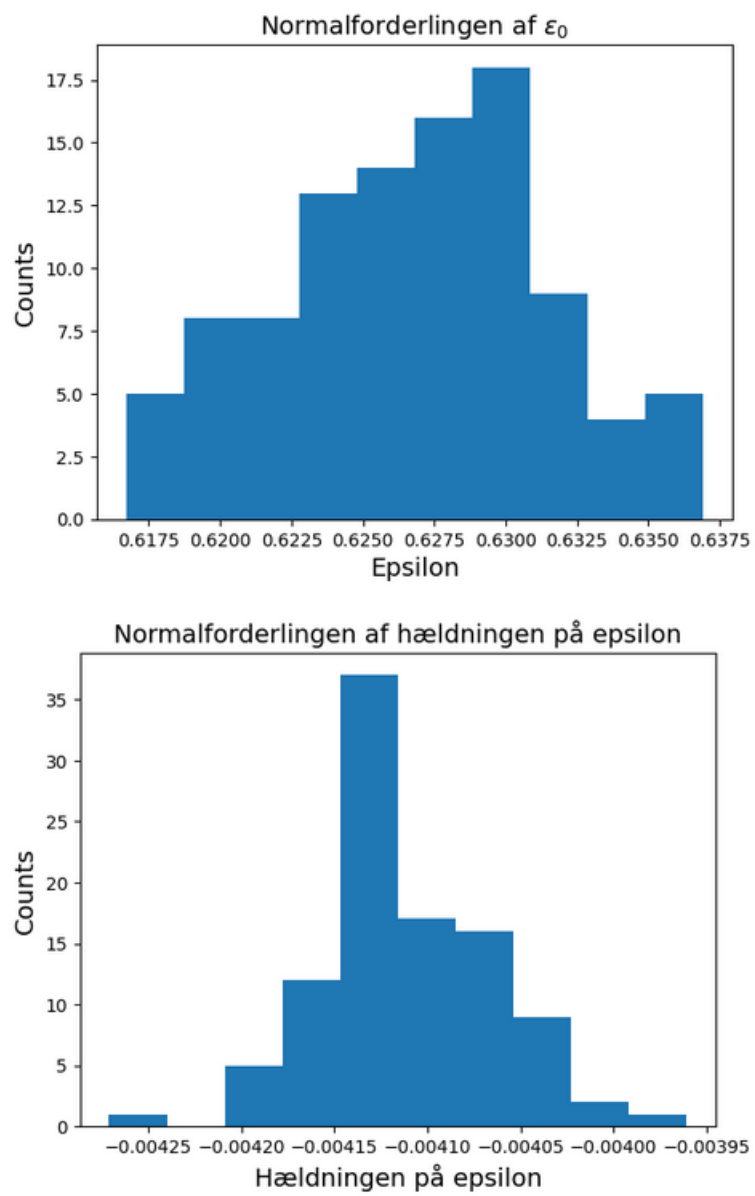
Normalfordelingen af starttemperaturen fra år 1850



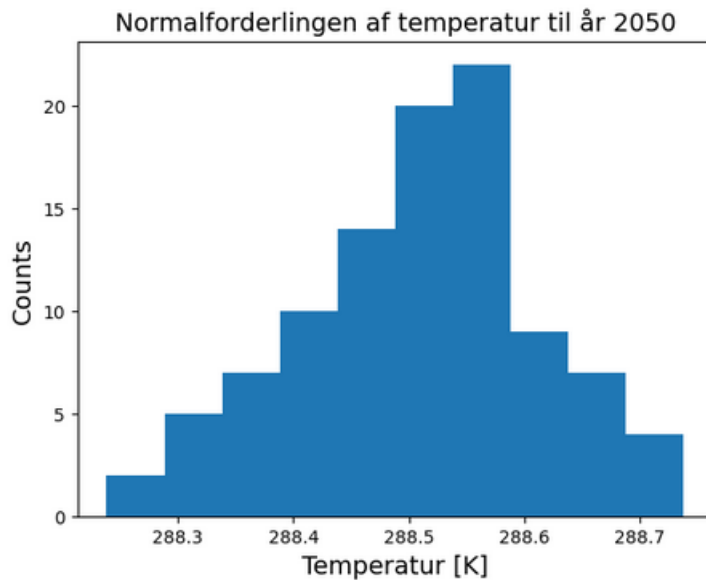
Figur 34



Figur 35



Figur 36



```
In [9]: #DEFINITION AF SENARIO: SSP1-1.9 med udvikling af solceller

#ESW = FS*(1-A(15))*np.pi*REarth**2*dt*secyear #Indkommende kortbølge-stråling beregnes
# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 1 # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fra det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = True #Tænd og sluk for udvikling af solceller
A_nr_1_on = False #Tænd og sluk for A1 eller A2

#TEMPERATUR BEREGNING
T102 = np.zeros(N)
EOcean102 = np.zeros(N)
C = m
eps0 = 0.6263779897
T0=286.8405
deps_dT=-0.00411
A0 = 0.297
T102[0]=T0 # Initialisering af temperatur
EOcean102[0]=T0/C # Initialisering af havets varme indhold

for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T102[i], EOcean102[i], t[i]])
    dRK4 = RK4(P)
    T102[i+1], EOcean102[i+1]= T102[i]+dRK4[0],EOcean102[i]+dRK4[1]
```

Figur 37

```

#USIKKERHEDS BEREGNING
B = 100
T12 = np.zeros((B,N))
EOcean12 = np.zeros(N)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dt_err = np.random.normal(deps_dT,deps_dt_err)

    T12[j,0]=T0 # Initialisering af temperatur
    EOcean12[0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T12[j,i], EOcean12[i], t[i]])
        dRK4 = RK4(P)
        T12[j,i+1], EOcean12[i+1]= T12[j,i]+dRK4[0],EOcean12[i]+dRK4[1]

U_T12 = np.zeros(N)
T12_up = np.zeros(N)
T12_down = np.zeros(N)

for i in range(N):
    U_T12[i] = np.std(T12[:,i])
    T12_up[i] = T102[i] + U_T12[i]
    T12_down[i] = T102[i] - U_T12[i]

```

```

In [10]: #PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

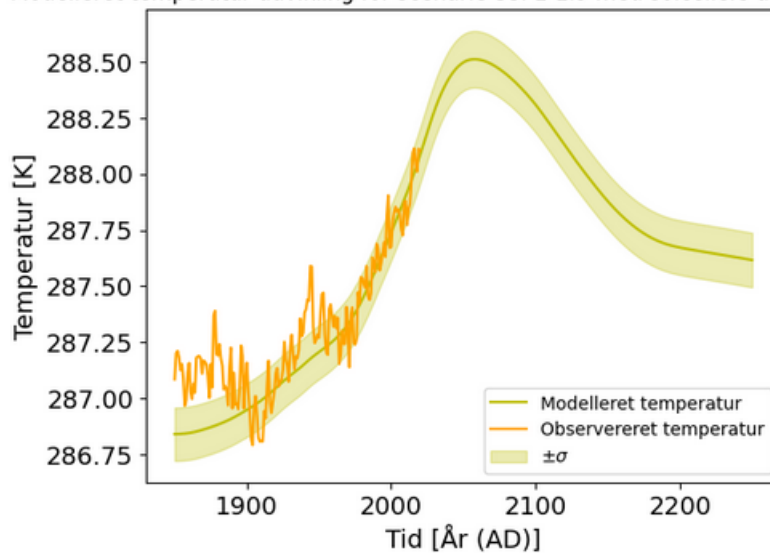
plt.plot(t,T102, label = "Modelleret temperatur", color = 'y')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T102[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t,T12_down,T12_up, label = "$\pm \sigma$", color = 'y', alpha=0.3)
plt.legend()
plt.title("Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 2")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

plt.show()
index = T102.argmax()
max_value12 = T102[index]
max_u12 = U_T12[index]
year = t[index]
print(T102[index]-T102[1523])
print(max_value12, max_u12, year)

```

Figur 38

Modelleret temperatur udvikling for scenarie SSP1-1.9 med solcellers albedo effect 2



0.7354408866816584
288.51113842337816 0.12577285084020481 2057.751937984496

```
In [11]: #DEFINITION AF SENARIO: SSP1-1.9 uden udvikling af solceller

# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 1 # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fra det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = False #Tænd og sluk for udvikling af solceller
A_nr_1_on = True #Tænd og sluk for A1 eller A2

#TEMPERATUR BEREGNING
T20 = np.zeros(N)
EOcean20 = np.zeros(N)
C = m
eps0 = 0.6263779897
deps_dT=-0.00411
T0=286.8405
A0 = 0.297
T20[0]=T0 # Initialisering af temperatur
EOcean20[0]=T0/m # Initialisering af havets varme indhold
```

Figur 39

```

for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T20[i], EOcean20[i], t[i]])
    dRK4 = RK4(P)
    T20[i+1], EOcean20[i+1] = T20[i]+dRK4[0],EOcean20[i]+dRK4[1]

#USIKKERHEDS BEREGNING
B = 100
T2 = np.zeros((B,N))
EOcean2 = np.zeros(N)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dt_err = np.random.normal(deps_dT,deps_dT_err)

    T2[j,0]=T0 # Initialisering af temperatur
    EOcean2[0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T2[j,i], EOcean2[i], t[i]])
        dRK4 = RK4(P)
        T2[j,i+1], EOcean2[i+1] = T2[j,i]+dRK4[0],EOcean2[i]+dRK4[1]

U_T2 = np.zeros(N)
T2_up = np.zeros(N)
T2_down = np.zeros(N)

for i in range(N):
    U_T2[i] = np.std(T2[:,i])
    T2_up[i] = T20[i] + U_T2[i]
    T2_down[i] = T20[i] - U_T2[i]

[12]: #PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

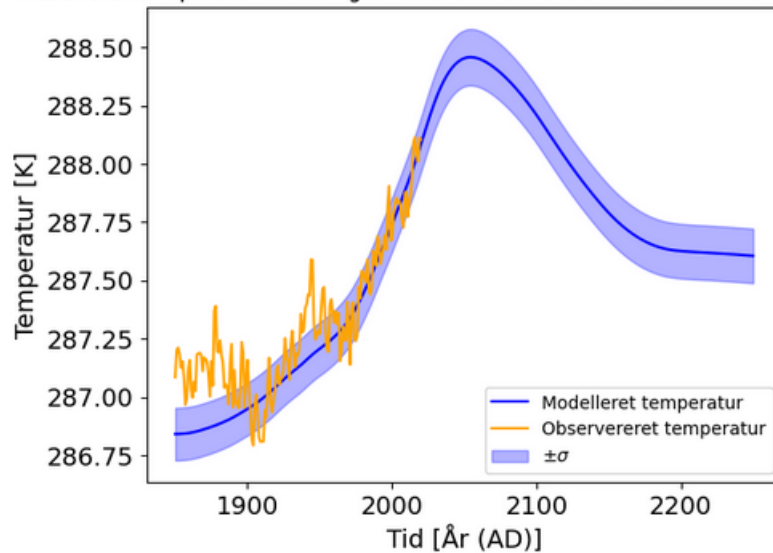
plt.plot(t,T20, label = "Modelleret temperatur", color = 'b')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T20[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t,T2_down,T2_up, label = "$\pm \sigma$", color = 'b', alpha=0.3)
plt.legend()
plt.title("Modelleret temperatur udvikling for scenarie SSP1-1.9 uden solcellers albedo effect")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

plt.show()
index = T20.argmax()
max_u2 = U_T2[index]
max_value2 = T20[index]
year = t[index]
print(T20[index]-T20[1523])
print(max_value2,max_u2, year)

```

Figur 40

Modelleret temperatur udvikling for scenarie SSP1-1.9 uden solcellers albedo effect



0.6815520201905656
288.45724955688706 0.12082268337596502 2054.351087771943

```
In [13]: N5=4000          # Antal tidsskridt
dt=1/10          # Tidsskridt-størrelse i år
t5 = np.linspace(t0, t0 + N5*dt, N5) #Tids-array

#DEFINITION AF SENARIO: SSP5-8.5 uden udvikling af solceller

# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 5          # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fro det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = False #Tænd og sluk for udvikling af solceller
A_nr_1_on = True #Tænd og sluk for A1 eller A2

#TEMPERATUR BEREGNING
T30 = np.zeros(N5)
EOcean20 = np.zeros(N5)
C = m
eps0 = 0.6263779897
T0=286.8405
deps_dT=-0.00411
A0 = 0.297
T30[0]=T0          # Initialisering af temperatur
EOcean20[0]=T0/C   # Initialisering af havets varme indhold
```

Figur 41

```

for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T30[i], EOcean20[i], t5[i]])
    dRK4 = RK4(P)
    T30[i+1], EOcean20[i+1]= T30[i]+dRK4[0],EOcean20[i]+dRK4[1]

#USIKKERHEDS BEREGNING
B = 100
T3 = np.zeros((B,N5))
EOcean2 = np.zeros(N5)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dT_err = np.random.normal(deps_dT,deps_dT_err)

    T3[j,0]=T0 # Initialisering af temperatur
    EOcean2[0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T3[j,i], EOcean2[i], t5[i]])
        dRK4 = RK4(P)
        T3[j,i+1], EOcean2[i+1]= T3[j,i]+dRK4[0],EOcean2[i]+dRK4[1]

U_T3 = np.zeros(N5)
T3_up = np.zeros(N5)
T3_down = np.zeros(N5)

for i in range(N5):
    U_T3[i] = np.std(T3[:,i])
    T3_up[i] = T30[i] + U_T3[i]
    T3_down[i] = T30[i] - U_T3[i]

#PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

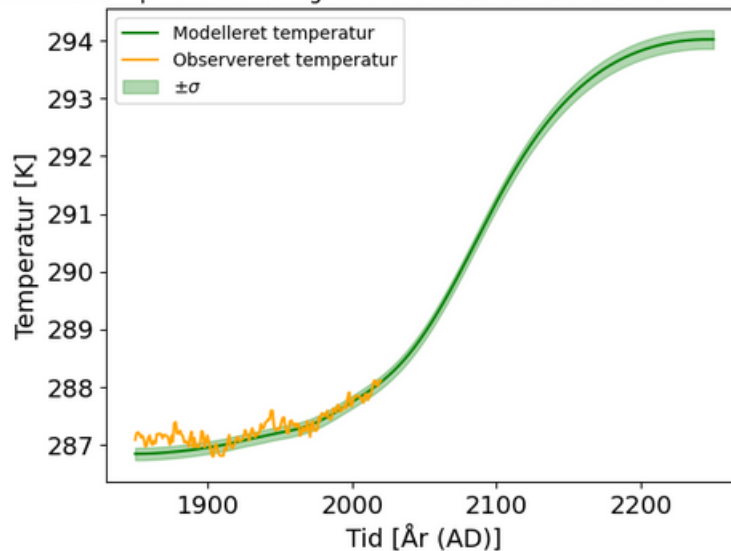
plt.plot(t5,T30, label = "Modelleret temperatur", color = 'g')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T30[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t5,T3_down,T3_up, label = "$\pm \sigma$", color = 'g', alpha=0.3)
plt.legend()
plt.title("Modelleret temperatur udvikling for scenarie SSP5-8.5 uden solcellers albedo effect")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

plt.show()
index = T30.argmax()
max_u3 = U_T3[index]
max_value3 = T30[index]
year = t[index]
print(T30[index]-T30[1515])
print(max_value3,max_u3, year)

```

Figur 42

Modelleret temperatur udvikling for scenarie SSP5-8.5 uden solcellers albedo effect



6.26027683402674
294.0232814847101 0.1592453599968548 2246.599149787447

```
In [15]: N5=4000          # Antal tidsskridt
dt=1/10         # Tidsskridt-størrelse i år
t5 = np.linspace(t0, t0 + N5*dt, N5) #Tids-array

#DEFINITION AF SENARIO: SSP5-8.5 uden udvikling af solceller

# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 4          # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fro det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = False #Tænd og sluk for udvikling af solceller
A_nr_1_on = True #Tænd og sluk for A1 eller A2

#TEMPERATUR BEREGNING
T302 = np.zeros(N5)
EOcean202 = np.zeros(N5)
C = m
eps0 = 0.6263779897
T0=286.8405
deps_dT=-0.00411
A0 = 0.297
T302[0]=T0          # Initialisering af temperatur
EOcean202[0]=T0/C   # Initialisering af havets varme indhold
```

Figur 43

```

for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T302[i], EOcean202[i], t5[i]])
    dRK4 = RK4(P)
    T302[i+1], EOcean202[i+1]= T302[i]+dRK4[0],EOcean202[i]+dRK4[1]

#USIKKERHEDS BEREGNING
B = 100
T32 = np.zeros((B,N5))
EOcean22 = np.zeros(N5)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dt_err = np.random.normal(deps_dT,deps_dT_err)

    T32[j,0]=T0 # Initialisering af temperatur
    EOcean22[j]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T32[j,i], EOcean22[i], t5[i]])
        dRK4 = RK4(P)
        T32[j,i+1], EOcean22[i+1]= T32[j,i]+dRK4[0],EOcean22[i]+dRK4[1]

U_T32 = np.zeros(N5)
T32_up = np.zeros(N5)
T32_down = np.zeros(N5)

for i in range(N5):
    U_T32[i] = np.std(T32[:,i])
    T32_up[i] = T302[i] + U_T32[i]
    T32_down[i] = T302[i] - U_T32[i]

```

```

n [48]: #PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

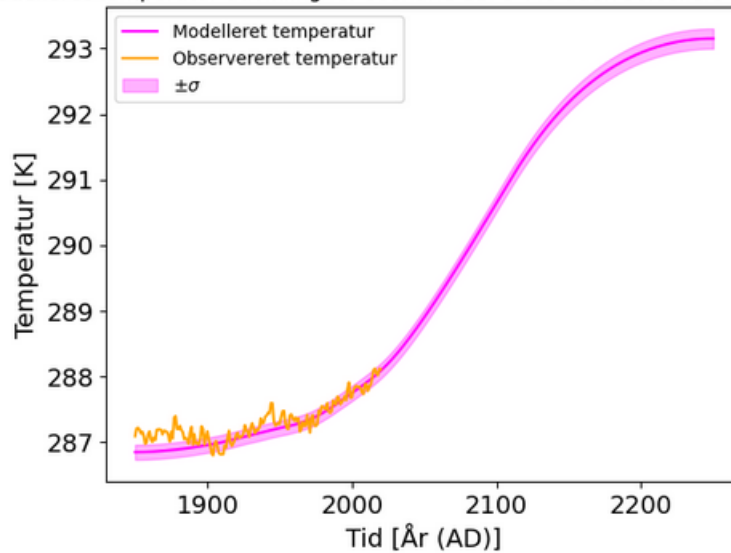
plt.plot(t5,T302, label = "Modelleret temperatur", color = 'magenta')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T302[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t5,T32_down,T32_up, label = "$\pm \sigma$", color = 'magenta', alpha=0.3)
plt.legend()
plt.title("Modelleret temperatur udvikling for scenarie SSP3-7.0 uden solcellers albedo effect")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

plt.show()
index = T302.argmax()
max_value32 = T302[index]
max_u32 = U_T32[index]
year = t[index]
print(T302[index]-T302[1515])
print(max_value32,max_u32, year)

```

Figur 44

Modelleret temperatur udvikling for scenarie SSP3-7.0 uden solcellers albedo effect



5.3880370586888375
293.1510417093722 0.1525577193301174 2249.099774943736

```
In [41]: N5=4000          # Antal tidsskridt
          dt=1/10        # Tidsskridt-størrelse i år
          t5 = np.linspace(t0, t0 + N5*dt, N5) #Tids-array

          #DEFINITION AF SENARIO: SSP5-8.5 uden udvikling af solceller

          # Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
          scenarios = 2          # Her vælges, hvilket SSP-scenarie man anvender!
          SSPint=interp1d(SSP[:,0],SSP[:,scenarios])          # Lineær interpolations-funktion defineres fra det valgte scenarie

          human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
          eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
          solcelle_vækst_on = False #Tænd og sluk for udvikling af solceller
          A_nr_1_on = True #Tænd og sluk for A1 eller A2

          #TEMPERATUR BEREGNING
          T22 = np.zeros(N5)
          EOcean22 = np.zeros(N5)
          C = m
          eps0 = 0.6263779897
          T0=286.8405
          deps_dT=-0.00411
          A0 = 0.297
          T22[0]=T0          # Initialisering af temperatur
          EOcean22[0]=T0/C    # Initialisering af havets varme indhold
```

Figur 45

```

for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T22[i], EOcean22[i], t5[i]])
    dRK4 = RK4(P)
    T22[i+1], EOcean22[i+1]= T22[i]+dRK4[0],EOcean22[i]+dRK4[1]

#USIKKERHEDS BEREGNING
B = 100
T220 = np.zeros((B,N5))
EOcean220 = np.zeros(N5)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (FS*(1-A0))/(4*SBconst*T0**4)
    deps_dt_err = np.random.normal(deps_dT,deps_dt_err)

    T220[j,0]=T0 # Initialisering af temperatur
    EOcean220[0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T220[j,i], EOcean220[i], t5[i]])
        dRK4 = RK4(P)
        T220[j,i+1], EOcean220[i+1]= T220[j,i]+dRK4[0],EOcean220[i]+dRK4[1]

U_T220 = np.zeros(N5)
T220_up = np.zeros(N5)
T220_down = np.zeros(N5)

for i in range(N5):
    U_T220[i] = np.std(T220[:,i])
    T220_up[i] = T22[i] + U_T220[i]
    T220_down[i] = T22[i] - U_T220[i]

```

```

[43]: N5=4000 # Antal tidsskridt
      dt=1/10 # Tidsskridt-størrelse i år
      t5 = np.linspace(t0, t0 + N5*dt, N5) #Tids-array

#DEFINITION AF SENARIO: SSP5-8.5 uden udvikling af solceller

# Interpolering af SSP drivhusgas-forceringsscenarier til de angivne tidsskridt
scenarios = 3 # Her vælges, hvilket SSP-scenarie man anvender!
SSPint=interp1d(SSP[:,0],SSP[:,scenarios]) # Lineær interpolations-funktion defineres fro det valgte scenarie

human_ghg_on = True # Tænd og sluk for menneskeskabt drivhusgas-forcering
eps_temp_dep = True # Tænd og sluk for temperatur afhængig emissivitet
solcelle_vækst_on = False #Tænd og sluk for udvikling af solceller
A_nr_1_on = True #Tænd og sluk for A1 eller A2

```

Figur 46

```

#TEMPERATUR BEREGNING
T320 = np.zeros(N5)
EOcean320 = np.zeros(N5)
C = m
eps0 = 0.6263779897
T0=286.8405
deps_dT=-0.00411
A0 = 0.297
T320[0]=T0 # Initialisering af temperatur
EOcean320[0]=T0/C # Initialisering af havets varme indhold

for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
    P = np.array([T320[i], EOcean320[i], t5[i]])
    dRK4 = RK4(P)
    T320[i+1], EOcean320[i+1]= T320[i]+dRK4[0],EOcean320[i]+dRK4[1]

#USIKKERHEDS BEREGNING
B = 100
T3200 = np.zeros((B,N5))
EOcean3200 = np.zeros(N5)

for j in range(B):
    C = m
    T0=286.8405
    eps0 = 0.6263779897
    deps_dT=-0.00411
    A0 = 0.297
    C = np.random.normal(m,m_err)
    A0 = np.random.normal(A0,A0_err)
    T0 = np.random.normal(T0,T0_err)
    #eps0 = np.random.normal(eps0,eps0_err)
    eps0 = (F5*(1-A0))/(4*SBconst*T0**4)
    deps_dT_err = np.random.normal(deps_dT,deps_dT_err)

    T3200[j,0]=T0 # Initialisering af temperatur
    EOcean3200[j,0]=T0/C # Initialisering af havets varme indhold

    for i in range(0,N5-1): #Løkke over alle tidsskridt "i" fra skridt 1 til skridt N
        P = np.array([T3200[j,i], EOcean3200[i], t5[i]])
        dRK4 = RK4(P)
        T3200[j,i+1], EOcean3200[i+1]= T3200[j,i]+dRK4[0],EOcean3200[i]+dRK4[1]

U_T3200 = np.zeros(N5)
T3200_up = np.zeros(N5)
T3200_down = np.zeros(N5)

for i in range(N5):
    U_T3200[i] = np.std(T3200[:,i])
    T3200_up[i] = T320[i] + U_T3200[i]
    T3200_down[i] = T320[i] - U_T3200[i]

9]: T0=286.8405
#PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur
plt.plot(t5,T30-T0, label = "SSP5-8.5", color = 'g')
plt.plot(t5,T302-T0, label = "SSP3-7.0", color = 'magenta')
plt.plot(t,T320-T0, label = "SSP2-4.5", color = 'y')
plt.plot(t,T22-T0, label = "SSP1-2.6", color = 'r')
plt.plot(t,T20-T0, label = "SSP1-1.9", color = 'b')
plt.fill_between(t5,T3_down-T0,T3_up-T0, label = "$\pm \sigma$", color = 'g', alpha=0.3)
plt.fill_between(t5,T32_down-T0,T32_up-T0, label = "$\pm \sigma$", color = 'magenta', alpha=0.3)

```

Figur 47

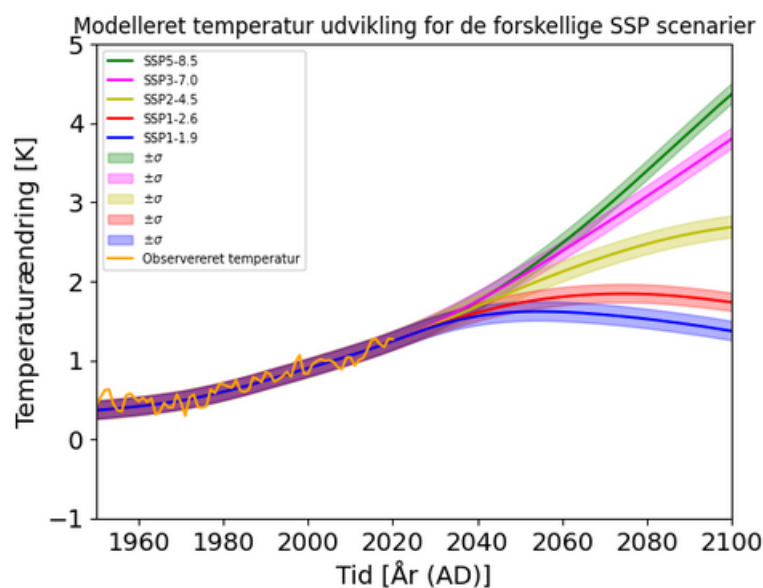
```

plt.fill_between(t5,T32_down-T0,T32_up-T0, label = "$\pm \sigma$", color = 'magenta', alpha=0.3)
plt.fill_between(t,T3200_down-T0,T3200_up-T0, label = "$\pm \sigma$", color = 'y', alpha=0.3)
plt.fill_between(t,T220_down-T0,T220_up-T0, label = "$\pm \sigma$", color = 'r', alpha=0.3)
plt.fill_between(t,T2_down-T0,T2_up-T0, label = "$\pm \sigma$", color = 'b', alpha=0.3)
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T10[11_00:14_00])-T0, label = "Observeret temperatur", color = 'orang')

plt.xlim(1950,2100)
plt.ylim(-1,5)
plt.legend(fontsize=7)
plt.title("Modelleret temperatur udvikling for de forskellige SSP scenarier")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperaturændring [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)

plt.show()
print(max_value1-max_value2)
print(max_value12-max_value2)
print(max_value3-max_value2)

```



```

0.07277148235061759
0.05388886649109281
5.566031927823019

```

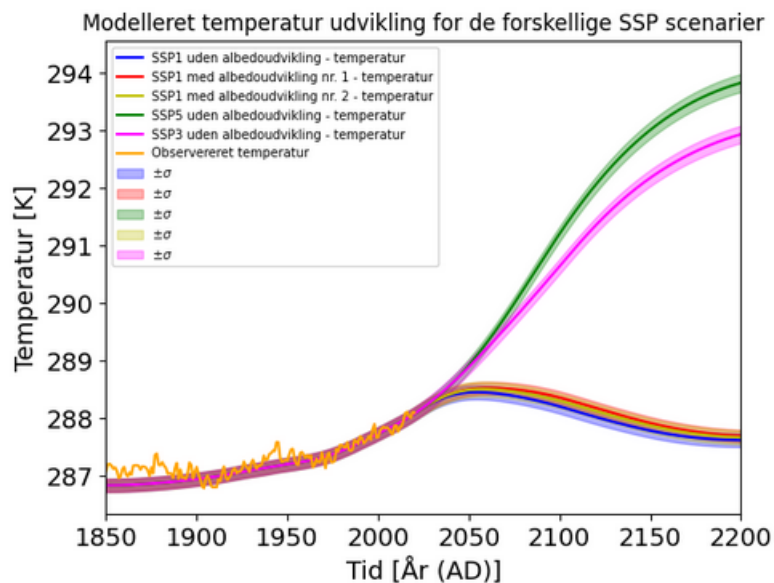
```

2]: plt.plot(t,T20, label = "SSP1 uden albedoudvikling - temperatur", color = 'b')
plt.plot(t,T10, label = "SSP1 med albedoudvikling nr. 1 - temperatur", color = 'r')
plt.plot(t,T102, label = "SSP1 med albedoudvikling nr. 2 - temperatur", color = 'y')
plt.plot(t5,T30, label = "SSP5 uden albedoudvikling - temperatur", color = 'g')

```

Figur 48

```
plt.plot(t5,T30, label = "SSP5 uden albedoudvikling - temperatur", color = 'g')
plt.plot(t5,T302, label = "SSP3 uden albedoudvikling - temperatur", color = 'magenta')
plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T10[11_00:14_00]), label = "Observeret temperatur", color = 'orange')
plt.fill_between(t,T2_down,T2_up, label = "$\pm \sigma$", color = 'b', alpha=0.3)
plt.fill_between(t,T1_down,T1_up, label = "$\pm \sigma$", color = 'r', alpha=0.3)
plt.fill_between(t5,T3_down,T3_up, label = "$\pm \sigma$", color = 'g', alpha=0.3)
plt.fill_between(t,T12_down,T12_up, label = "$\pm \sigma$", color = 'y', alpha=0.3)
plt.fill_between(t5,T32_down,T32_up, label = "$\pm \sigma$", color = 'magenta', alpha=0.3)
plt.xlim(1850,2200)
plt.legend(fontsize=7)
plt.title("Modelleret temperatur udvikling for de forskellige SSP scenarier")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)
```



```
|: #PLOTNING
ObsOceanT=np.loadtxt('HadSST4_ocean_annual.txt') # Indlæsning af observeret havtemperatur

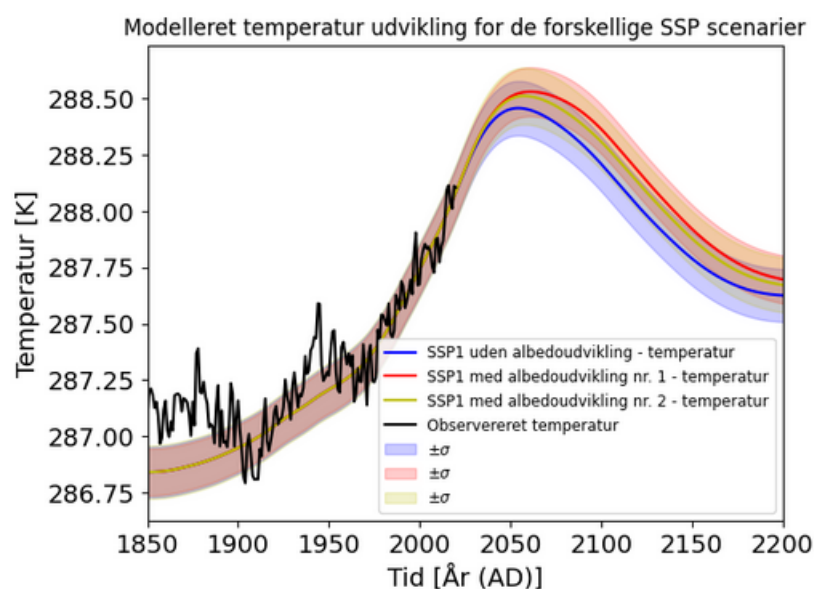
plt.plot(t,T20, label = "SSP1 uden albedoudvikling - temperatur", color = 'b')
plt.plot(t,T10, label = "SSP1 med albedoudvikling nr. 1 - temperatur", color = 'r')
plt.plot(t,T102, label = "SSP1 med albedoudvikling nr. 2 - temperatur", color = 'y')

plt.plot(ObsOceanT[:,0],ObsOceanT[:,1]+np.mean(T10[11_00:14_00]), label = "Observeret temperatur", color = 'black')
plt.fill_between(t,T2_down,T2_up, label = "$\pm \sigma$", color = 'b', alpha=0.2)
plt.fill_between(t,T1_down,T1_up, label = "$\pm \sigma$", color = 'r', alpha=0.2)
plt.fill_between(t,T12_down,T12_up, label = "$\pm \sigma$", color = 'y', alpha=0.2)
plt.legend(fontsize=8.5)
plt.title("Modelleret temperatur udvikling for de forskellige SSP scenarier")
plt.xlabel('Tid [År (AD)]',fontsize=14)
plt.ylabel('Temperatur [K]',fontsize=14)
plt.tick_params(which='major',labelsize=14)
```

Figur 49

```
plt.ylabel('Temperatur [K]', fontsize=14)
plt.tick_params(which='major', labels=14)
plt.xlim(1850, 2200)

plt.show()
print(max_value1, max_value12, max_value3)
print(max_value1-max_value2)
print(max_value12-max_value2)
```



```
288.5300210392377 288.51113842337816 294.0232814847101
0.07277148235061759
0.05388886649109281
```

1:

Figur 50