import numpy as np import sympy as sp from sympy import symbols import matplotlib as plt x, k, n = sp.symbols('x, k, n') def prod(k,n):C\_s\_XK=kost(N\_sol,k) C\_s\_XKC\_s\_fn = sp.lambdify( (x,k), C\_s\_XK ) C\_s\_fn(x,k) return (n\*\*0.5)\*(k\*\*0.5) def kost(n,k): return 350\*n+(500\*k)+10000 N\_sol = sp.solve(prod(k,n) - x, n)[0] N\_sol

**ChatGPT sa:**

You're working on a symbolic computation using sympy, but the code you've posted has several issues that will prevent it from running correctly. Here's a breakdown of what needs fixing and a corrected version of your script:

**Issues in your code:**

1. **Syntax errors:**
   * The prod function is missing a proper function body; the return statement is after another block.
   * The call to kost(N\_sol, k) is made before kost is defined or N\_sol is computed.
   * C\_s\_fn(x,k) is not a valid statement (likely a misplaced call).
   * import matplotlib as plt is unnecessary here (and incorrect; should be import matplotlib.pyplot as plt if needed).
2. **Logical structure:**
   * You're trying to define a cost function kost(n, k) and use it in a production function prod(k, n) to solve for n symbolically.
   * However, you're calling kost inside prod, and prod is used to solve for n **before** you define kost.

**Corrected and functional version:**

python

KopierRediger

import sympy as sp

# Define symbols

x, k, n = sp.symbols('x k n')

# Define the cost function

def kost(n, k):

return 350\*n + 500\*k + 10000

# Define the production function

def prod(k, n):

return sp.sqrt(n) \* sp.sqrt(k)

# Now solve prod(k, n) = x for n

# This means: sqrt(n) \* sqrt(k) = x -> sqrt(n) = x / sqrt(k) -> n = (x / sqrt(k))\*\*2

solution\_n = sp.solve(prod(k, n) - x, n)[0]

# Print solution

print("Solution for n in terms of x and k:")

sp.pprint(solution\_n)

# Optionally: substitute into cost function

cost\_expr = kost(solution\_n, k)

print("\nCost as a function of x and k:")

sp.pprint(cost\_expr)

**Sample output (pretty printed):**

javascript

KopierRediger

Solution for n in terms of x and k:

2

x²

──────

k

Cost as a function of x and k:

2

350⋅x

────── + 500⋅k + 10000

k

Let me know if you're trying to optimize this cost or plot it—happy to take it further!

4o

**Du sa:**

import numpy as np import sympy as sp from sympy import symbols import matplotlib as plt x, k, n = sp.symbols('x, k, n') def prod(k,n): return (n\*\*0.5)\*(k\*\*0.5) def kost(n,k): return 350\*n+(500\*k)+10000 N\_sol = sp.solve(prod(k,n) - x, n)[0] N\_sol C\_s\_XK=kost(N\_sol,k) C\_s\_XKC\_s\_fn = sp.lambdify( (x,k), import matplotlib.pyplot as plt p\_num=np.linspace(0.1,10,100) fig, ax = plt.subplots() ax.set\_ylabel('Overskudd', loc='top', rotation=0) # Label on top, no rotation ax.set\_xlabel('Enheter arbeidskraft (N)', loc='right') # Label on right ax.set\_xlim(0, 10) # Set x-axis range ax.set\_ylim(0, 600) # Set y-axis range # Remove the top and right spines for a cleaner look ax.spines['top'].set\_color('none') ax.spines['right'].set\_color('none') ax.plot(p\_num, C\_s\_fn(x,10), label='overskudd') plt.show() C\_s\_XK ) C\_s\_fn(x,k)

**ChatGPT sa:**

Thanks for sharing your updated code! You're close, but there are several syntax and structural issues that will cause errors. Let me help you clean this up and get a working version.

**✅ What You're Trying to Do:**

* Define a production function prod(k, n).
* Solve for n in terms of x and k.
* Substitute that into a cost function kost(n, k).
* Convert the symbolic expression into a numerical function using lambdify.
* Plot the cost as a function of x for a fixed k.

**❌ Problems in Your Code:**

1. You accidentally split the lambdify call and inserted import matplotlib.pyplot as plt in the middle.
2. You're calling C\_s\_fn(x, 10) inside ax.plot() before C\_s\_fn is defined.
3. You're using x symbolically, but in your plot, it needs to be a numpy array (p\_num).
4. The label 'overskudd' doesn't match the plotted values (should be cost or profit?).

**✅ Fixed Version:**

python

KopierRediger

import numpy as np

import sympy as sp

import matplotlib.pyplot as plt

# Define symbolic variables

x, k, n = sp.symbols('x k n')

# Define production and cost functions

def prod(k, n):

return (n\*\*0.5)\*(k\*\*0.5)

def kost(n, k):

return 350\*n + 500\*k + 10000

# Solve for n in terms of x and k from production function

N\_sol = sp.solve(prod(k, n) - x, n)[0]

# Substitute N\_sol into cost function to get cost as function of x and k

C\_s\_XK = kost(N\_sol, k)

# Lambdify for numerical evaluation

C\_s\_fn = sp.lambdify((x, k), C\_s\_XK)

# Generate x values (e.g. production levels)

x\_vals = np.linspace(0.1, 10, 100)

k\_fixed = 10 # fixed capital value

# Evaluate cost for each x

cost\_vals = C\_s\_fn(x\_vals, k\_fixed)

# Plot

fig, ax = plt.subplots()

ax.set\_ylabel('Kostnad', loc='top', rotation=0)

ax.set\_xlabel('Produksjon (x)', loc='right')

ax.set\_xlim(0, 10)

ax.set\_ylim(0, max(cost\_vals) \* 1.1)

# Clean up spines

ax.spines['top'].set\_color('none')

ax.spines['right'].set\_color('none')

# Plot the cost function

ax.plot(x\_vals, cost\_vals, label='Totalkostnad', color='blue')

ax.legend()

plt.title('Totalkostnad som funksjon av produksjon (x)')

plt.grid(True)

plt.show()

**🔍 Notes:**

* x\_vals is your production quantity.
* You're plotting total cost as a function of x, with k held constant.
* You can modify the code to plot for different values of k or even plot profit if you subtract cost from revenue.

Let me know if you'd like to add revenue or profit to this graph too!

4o

**Du sa:**

**ChatGPT sa:**

Takk for bildet! Oppgaven er tredelt og handler om kostnadsanalyse og skalafordeler for en produksjonsfunksjon på kort sikt. La oss gå gjennom del for del.

**🔍 Oppgavetekst Oppsummert**

**a)**

* Finn uttrykk for total produksjonskostnad Cs(x,K0)C\_s(x, K\_0)Cs​(x,K0​) som funksjon av produsert mengde xxx og gitt kapitalmengde K0K\_0K0​.
* Tegn dette uttrykket grafisk for én K0K\_0K0​, og for en annen kapitalmengde K1=50K\_1 = 50K1​=50.
* Kommenter funnene dine.

**b)**

* Tegn:
  + Totale gjennomsnittskostnader (ATC),
  + Variable gjennomsnittskostnader (AVC),
  + Grensekostnad (MC).
* Kommenter og forklar forholdet mellom disse.

**c)**

* Hvilke **skalaegenskaper** viser denne produksjonsfunksjonen?

**✅ Løsning Del a: Total Kostnad Cs(x,K)C\_s(x, K)Cs​(x,K)**

**Produksjonsfunksjon:**

x=n⋅k⇒n=(xk)2=x2kx = \sqrt{n} \cdot \sqrt{k} \Rightarrow n = \left( \frac{x}{\sqrt{k}} \right)^2 = \frac{x^2}{k}x=n​⋅k​⇒n=(k​x​)2=kx2​

**Kostnadsfunksjon:**

Cs(x,K)=350n+500K+10 000=350⋅x2K+500K+10 000C\_s(x, K) = 350n + 500K + 10\,000 = 350 \cdot \frac{x^2}{K} + 500K + 10\,000Cs​(x,K)=350n+500K+10000=350⋅Kx2​+500K+10000

Så:

Cs(x,K)=350x2K+500K+10 000C\_s(x, K) = \frac{350x^2}{K} + 500K + 10\,000Cs​(x,K)=K350x2​+500K+10000

**🔢 Plotting**

Vi kan bruke koden din som utgangspunkt og plotte kostnadsfunksjonen for to ulike kapitalnivåer: K0=10K\_0 = 10K0​=10 og K1=50K\_1 = 50K1​=50.

**🧮 Kode for Del a – Plotting av Cs(x,K)C\_s(x, K)Cs​(x,K)**

python

KopierRediger

import numpy as np

import matplotlib.pyplot as plt

# Parametre

K\_0 = 10

K\_1 = 50

x\_vals = np.linspace(0.1, 20, 200)

# Kostnadsfunksjon

def C\_s(x, K):

return (350 \* x\*\*2) / K + 500 \* K + 10000

# Verdier

C\_vals\_K0 = C\_s(x\_vals, K\_0)

C\_vals\_K1 = C\_s(x\_vals, K\_1)

# Plot

plt.figure(figsize=(8, 5))

plt.plot(x\_vals, C\_vals\_K0, label=r'$C\_s(x, K\_0=10)$')

plt.plot(x\_vals, C\_vals\_K1, label=r'$C\_s(x, K\_1=50)$', linestyle='--')

plt.xlabel('Produksjonsmengde (x)')

plt.ylabel('Totale kostnader')

plt.title('Totale kostnader ved ulik kapitalmengde')

plt.legend()

plt.grid(True)

plt.show()

**💬 Kommentarer til funn:**

* Når kapitalmengden øker (fra K0=10K\_0 = 10K0​=10 til K1=50K\_1 = 50K1​=50), synker den variable delen av kostnaden fordi færre arbeidere trengs.
* Faste kostnader (kapitalkostnader) øker, men total kostnad vokser **langsommere** med produksjon når KKK er høyere.
* Illustrerer **trade-off** mellom arbeidskraft og kapital i produksjonsprosessen.

Vil du at jeg skal gå videre med del **b** og lage grafer for ATC, AVC og MC?

**Du sa:**

kunne du løst oppgave a med sympy

**ChatGPT sa:**

Selvfølgelig! La oss løse **oppgave a** med sympy slik at du får det symbolsk – inkludert uttrykket for total produksjonskostnad Cs(x,K)C\_s(x, K)Cs​(x,K), og vi kan også gjøre plotting for to verdier av kapitalmengde K0=10K\_0 = 10K0​=10 og K1=50K\_1 = 50K1​=50.

**🔢 Symbolsk løsning med sympy**

**🎯 Mål:**

1. **Uttrykk:** Finn total kostnad Cs(x,K)C\_s(x, K)Cs​(x,K) som funksjon av produsert mengde og kapital.
2. **Plot:** Tegn for to kapitalmengder.

**✅ Kode i sympy:**

python

KopierRediger

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

# Definer symbolene

x, k, n = sp.symbols('x k n')

# Produksjonsfunksjon: x = sqrt(n) \* sqrt(k)

# Løs for n: x = sqrt(nk) -> n = x^2 / k

n\_expr = x\*\*2 / k

# Kostnadsfunksjon: C = 350n + 500k + 10000

C\_expr = 350 \* n\_expr + 500 \* k + 10000

# Forenkle uttrykket

C\_expr\_simplified = sp.simplify(C\_expr)

print("Symbolsk uttrykk for C\_s(x, k):")

sp.pprint(C\_expr\_simplified)

# Lag numerisk funksjon

C\_fn = sp.lambdify((x, k), C\_expr\_simplified, modules='numpy')

# Verdier for plotting

x\_vals = np.linspace(0.1, 20, 200)

K0 = 10

K1 = 50

C\_vals\_K0 = C\_fn(x\_vals, K0)

C\_vals\_K1 = C\_fn(x\_vals, K1)

# Plot

plt.figure(figsize=(8, 5))

plt.plot(x\_vals, C\_vals\_K0, label=r'$C\_s(x, K\_0=10)$', color='blue')

plt.plot(x\_vals, C\_vals\_K1, label=r'$C\_s(x, K\_1=50)$', color='green', linestyle='--')

plt.xlabel('Produksjon (x)')

plt.ylabel('Totale kostnader')

plt.title('Totale kostnader ved ulike kapitalmengder')

plt.legend()

plt.grid(True)

plt.show()

**📌 Resultat**

**Symbolsk uttrykk:**

Cs(x,k)=350x2k+500k+10000C\_s(x, k) = \frac{350x^2}{k} + 500k + 10000Cs​(x,k)=k350x2​+500k+10000

**💬 Tolkning:**

* Når kkk øker, synker 350x2k\frac{350x^2}{k}k350x2​, altså den variable delen av kostnaden → krever mindre arbeidskraft.
* Samtidig øker den faste delen 500k500k500k (kapitalkostnader).
* Høyere kapital gir lavere marginalkostnader for høyere produksjon.

Klar for å gå videre til **oppgave b** og se på gjennomsnittskostnader og grensekostnader?

**Du sa:**

Hvordan løser man oppgave b med sympy?

**ChatGPT sa:**

Flott spørsmål! I **oppgave b** skal vi bruke det vi fant i oppgave a til å:

1. **Finne uttrykk for**:
   * Totale gjennomsnittskostnader (ATC = C(x)/x),
   * Variable gjennomsnittskostnader (AVC),
   * Grensekostnader (MC = dC/dx),
2. **Tegne dem** i én figur.
3. **Kommentere forholdet mellom dem.**

Vi gjør dette symbolsk med sympy, og deretter gjør vi tallfesting og plotting med numpy og matplotlib.

**📘 Teori**

Fra forrige del hadde vi kostnadsfunksjonen:

Cs(x,k)=350x2k+500k+10000C\_s(x, k) = \frac{350x^2}{k} + 500k + 10000Cs​(x,k)=k350x2​+500k+10000

La oss nå definere:

* **Totale gjennomsnittskostnader (ATC)**:

ATC(x,k)=Cs(x,k)xATC(x, k) = \frac{C\_s(x, k)}{x}ATC(x,k)=xCs​(x,k)​

* **Variable gjennomsnittskostnader (AVC)**:

AVC(x,k)=VC(x,k)x=350x2/kx=350xkAVC(x, k) = \frac{VC(x, k)}{x} = \frac{350x^2/k}{x} = \frac{350x}{k}AVC(x,k)=xVC(x,k)​=x350x2/k​=k350x​

* **Grensekostnader (MC)**:

MC(x,k)=dCs(x,k)dxMC(x, k) = \frac{dC\_s(x, k)}{dx}MC(x,k)=dxdCs​(x,k)​

**✅ Løsning i sympy (og plott i matplotlib)**

python

KopierRediger

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

# Symboler

x, k = sp.symbols('x k')

# Total kostnad fra del a

C\_expr = (350 \* x\*\*2) / k + 500 \* k + 10000

# Gjennomsnittskostnader

ATC\_expr = sp.simplify(C\_expr / x)

AVC\_expr = sp.simplify((350 \* x\*\*2 / k) / x) # kun variable kostnader

MC\_expr = sp.simplify(sp.diff(C\_expr, x)) # grensekostnad

# Vis symbolsk

print("\nTotale gjennomsnittskostnader (ATC):")

sp.pprint(ATC\_expr)

print("\nVariable gjennomsnittskostnader (AVC):")

sp.pprint(AVC\_expr)

print("\nGrensekostnader (MC):")

sp.pprint(MC\_expr)

# Numeriske funksjoner

ATC\_fn = sp.lambdify((x, k), ATC\_expr, 'numpy')

AVC\_fn = sp.lambdify((x, k), AVC\_expr, 'numpy')

MC\_fn = sp.lambdify((x, k), MC\_expr, 'numpy')

# Verdier for plotting

x\_vals = np.linspace(0.1, 20, 300)

k\_val = 10

# Evaluer funksjonene

ATC\_vals = ATC\_fn(x\_vals, k\_val)

AVC\_vals = AVC\_fn(x\_vals, k\_val)

MC\_vals = MC\_fn(x\_vals, k\_val)

# Plot

plt.figure(figsize=(8,5))

plt.plot(x\_vals, ATC\_vals, label='ATC (Total gj.snittskostnad)', linewidth=2)

plt.plot(x\_vals, AVC\_vals, label='AVC (Variabel gj.snittskostnad)', linestyle='--')

plt.plot(x\_vals, MC\_vals, label='MC (Grensekostnad)', linestyle='-.')

plt.xlabel('Produksjonsmengde (x)')

plt.ylabel('Kostnad per enhet')

plt.title(f'Kostnadsfunksjoner ved k = {k\_val}')

plt.legend()

plt.grid(True)

plt.show()

**💬 Kommentar til forholdet mellom ATC, AVC og MC:**

* **MC krysser både AVC og ATC i deres minimumspunkt.** Dette er en klassisk økonomisk egenskap.
* **AVC < ATC** alltid, fordi ATC inkluderer faste kostnader.
* Når **MC < ATC**, trekker den ATC ned. Når **MC > ATC**, drar den ATC opp.
* Samme forhold gjelder for MC og AVC.

Vil du at vi også tolker dette grafisk eller går videre til oppgave **c** om skalafordeler?

**Du sa:**

Grafen til oppgave b skal se slik ut.

**ChatGPT sa:**

Perfekt! Takk for bildet – det viser akkurat hvordan grafen i **oppgave b** skal se ut. Her er hva vi observerer og hvordan du kan reprodusere det med korrekt kode og matematiske uttrykk i sympy.

**✅ Mål: Gjenskape denne grafen (ATC, AVC, MC) med sympy og matplotlib**

**Basert på bildet:**

* Kapitalmengden er satt til K=10K = 10K=10
* xxx går opp til 100
* MC (grønn linje) stiger jevnt lineært (den er MC(x)=700x/kMC(x) = 700x/kMC(x)=700x/k)
* AVC og ATC synker først, har minimum og så øker (klassisk U-form)

**🔢 Symbolsk løsning og kode**

Vi bruker:

Cs(x,k)=350x2k+500k+10000C\_s(x, k) = \frac{350x^2}{k} + 500k + 10000Cs​(x,k)=k350x2​+500k+10000

**Utledet:**

* ATC=C(x)x=350xk+500k+10000xATC = \frac{C(x)}{x} = \frac{350x}{k} + \frac{500k + 10000}{x}ATC=xC(x)​=k350x​+x500k+10000​
* AVC=350xkAVC = \frac{350x}{k}AVC=k350x​
* MC=dCdx=700xkMC = \frac{dC}{dx} = \frac{700x}{k}MC=dxdC​=k700x​

**✅ Kode som gir samme graf:**

python

KopierRediger

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

# Symboler

x, k = sp.symbols('x k')

# Kostnadsuttrykk

C\_expr = (350 \* x\*\*2) / k + 500 \* k + 10000

# ATC, AVC, MC

ATC\_expr = C\_expr / x

AVC\_expr = (350 \* x\*\*2 / k) / x # bare variable kostnader / x

MC\_expr = sp.diff(C\_expr, x)

# Lambdify

ATC\_fn = sp.lambdify((x, k), ATC\_expr, modules='numpy')

AVC\_fn = sp.lambdify((x, k), AVC\_expr, modules='numpy')

MC\_fn = sp.lambdify((x, k), MC\_expr, modules='numpy')

# Verdier

x\_vals = np.linspace(1, 100, 300)

k\_val = 10

atc\_vals = ATC\_fn(x\_vals, k\_val)

avc\_vals = AVC\_fn(x\_vals, k\_val)

mc\_vals = MC\_fn(x\_vals, k\_val)

# Plot

plt.figure(figsize=(10,6))

plt.plot(x\_vals, atc\_vals, label='totale gjennomsnittskostnad', color='blue')

plt.plot(x\_vals, avc\_vals, label='variable gjennomsnittskostnad', color='orange')

plt.plot(x\_vals, mc\_vals, label='grensekostnad', color='green')

plt.title('Gjennomsnittskostnad og grensekostnad på kort sikt (K=10)')

plt.xlabel('Produksjonsmengde (x)')

plt.ylabel('Kostnad per enhet')

plt.grid(True)

plt.legend()

plt.ylim(0, 11000)

plt.xlim(0, 100)

plt.show()

**📌 Kommentar til formen:**

* **AVC:** Stiger lineært med xxx, siden AVC=350xk\text{AVC} = \frac{350x}{k}AVC=k350x​
* **ATC:** Har U-form pga faste kostnader fordeles over flere enheter
* **MC:** Lineær funksjon som starter i null: MC=700xk\text{MC} = \frac{700x}{k}MC=k700x​

Vil du at jeg også skal skrive kommentarene som passer til grafen i oppgaven, slik du kan bruke dem direkte i besvarelsen?

**Du sa:**

Det visker som om de variable gjennomsnittskostnadene ikke blir riktig

**ChatGPT sa:**

Godt observert! Det ser ut som den **variable gjennomsnittskostnaden (AVC)** i grafen din ikke følger den riktige formen – den skal være en rett linje som øker jevnt, siden vi vet at:

**📘 Teori: Variabel gjennomsnittskostnad (AVC)**

Fra produksjonsfunksjonen og kostnadsmodellen:

VC(x,k)=350x2k⇒AVC(x,k)=VCx=350xkVC(x, k) = \frac{350x^2}{k} \quad \Rightarrow \quad AVC(x, k) = \frac{VC}{x} = \frac{350x}{k}VC(x,k)=k350x2​⇒AVC(x,k)=xVC​=k350x​

Som betyr at:

* AVC er **lineær** i xxx
* Den stiger jevnt (ikke krummer som ATC)

**❌ Vanlig feil:**

Du har kanskje brukt **hele kostnadsfunksjonen** C(x)C(x)C(x) for å lage både ATC og AVC, noe som gir samme kurveform for begge – men **AVC skal kun ta med den variable delen** (altså uten 500k og 10000).

**✅ Riktig kode for AVC**

Slik bør AVC\_expr defineres i sympy:

python

KopierRediger

AVC\_expr = sp.simplify((350 \* x) / k)

Eller direkte uten å gå via VCVCVC:

python

KopierRediger

AVC\_expr = 350 \* x / k

Og så lambdify:

python

KopierRediger

AVC\_fn = sp.lambdify((x, k), AVC\_expr, modules='numpy')

**🔁 Rask fiks i koden:**

**Bytt ut denne linjen**:

python

KopierRediger

AVC\_expr = (350 \* x\*\*2 / k) / x

**Med denne**:

python

KopierRediger

AVC\_expr = 350 \* x / k

Det vil gjøre at AVC oppfører seg som den rette linjen i figuren du tok bilde av 🎯

Vil du at jeg skal vise full oppdatert kode for hele figuren med riktig AVC?

4o

**Du sa:**

Kunne du gjort oppgave a med sympy og ax.plot istedenfor plt.plot, definer også produksjonen og bruk den til å løs oppgaven, kunne også definert utrykkene som funksjoner?

3/3

**ChatGPT sa:**

Absolutt! La oss gjøre **oppgave a** fra bunnen av, med fokus på:

**✅ Hva vi skal gjøre:**

1. **Definere produksjonsfunksjonen** symbolsk i sympy
2. **Løse for arbeidskraft nnn** symbolsk
3. **Bruke dette til å finne total kostnad Cs(x,k)C\_s(x, k)Cs​(x,k)** som en funksjon
4. **Gjøre det numerisk** med lambdify
5. **Bruke ax.plot()** med matplotlib for å tegne kostnadsfunksjonen for to ulike kapitalmengder: K0=10K\_0 = 10K0​=10 og K1=50K\_1 = 50K1​=50
6. **Alt organisert i definerte funksjoner**

**✅ Full kode med sympy, ax.plot() og funksjonsstruktur**

python

KopierRediger

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

# -----------------------------

# Symbolsk del

# -----------------------------

# 1. Symboler

x, k, n = sp.symbols('x k n')

# 2. Produksjonsfunksjon: x = sqrt(n) \* sqrt(k) = sqrt(nk)

def produksjon(n, k):

return sp.sqrt(n \* k)

# 3. Løs for n: x = sqrt(nk) -> n = x^2 / k

n\_expr = sp.solve(produksjon(n, k) - x, n)[0]

# 4. Kostnadsfunksjon: C = 350n + 500k + 10000

def kostnad(n, k):

return 350 \* n + 500 \* k + 10000

# 5. Sett inn n(x, k) → C\_s(x, k)

C\_expr = kostnad(n\_expr, k)

C\_expr = sp.simplify(C\_expr)

# 6. Gjør den numerisk

C\_fn = sp.lambdify((x, k), C\_expr, modules='numpy')

# -----------------------------

# Plotting med ax.plot()

# -----------------------------

# 7. Definer produksjonsmengder og kapitalnivåer

x\_vals = np.linspace(0.1, 20, 200)

K0 = 10

K1 = 50

# 8. Beregn kostnader

C\_vals\_K0 = C\_fn(x\_vals, K0)

C\_vals\_K1 = C\_fn(x\_vals, K1)

# 9. Plot med ax-objekt

fig, ax = plt.subplots(figsize=(8, 5))

ax.plot(x\_vals, C\_vals\_K0, label=r'$C\_s(x, K\_0=10)$', color='blue')

ax.plot(x\_vals, C\_vals\_K1, label=r'$C\_s(x, K\_1=50)$', color='green', linestyle='--')

# 10. Akseoppsett og stil

ax.set\_xlabel('Produksjonsmengde (x)', loc='right')

ax.set\_ylabel('Totale kostnader', loc='top', rotation=0)

ax.set\_title('Totale kostnader ved ulike kapitalmengder')

ax.spines['top'].set\_visible(False)

ax.spines['right'].set\_visible(False)

ax.grid(True)

ax.legend()

plt.tight\_layout()

plt.show()

**✨ Symbolsk kostnadsuttrykk**

Etter forenkling ser det slik ut:

Cs(x,k)=350x2k+500k+10000C\_s(x, k) = \frac{350x^2}{k} + 500k + 10000Cs​(x,k)=k350x2​+500k+10000

**💬 Tolkning (som svar på oppgavetekst a)**

* Ved høyere kapital (K1=50K\_1 = 50K1​=50) reduseres den variable delen av kostnaden, fordi man trenger færre arbeidere for å produsere samme mengde.
* Likevel øker de faste kostnadene (kapitalkostnad), som gir høyere kostnad ved lav produksjon.
* Ved høy produksjon er det mer lønnsomt med høy kapital.

Vil du jeg skal utvide med tekstkommentarene som passer i en innlevering?

4o

**Du sa:**

bruk funksjonene fra oppgave a og løs oppgave b på samme måte

**ChatGPT sa:**

Perfekt! Nå skal vi bygge videre på **oppgave a**, og bruke samme struktur – med sympy, ax.plot(), og funksjoner – for å løse **oppgave b**:

**✅ Hva krever oppgave b?**

Vi skal:

1. Bruke kostnadsfunksjonen Cs(x,k)C\_s(x, k)Cs​(x,k) fra oppgave a
2. Lage uttrykk og funksjoner for:
   * Totale gjennomsnittskostnader (ATC)
   * Variable gjennomsnittskostnader (AVC)
   * Grensekostnad (MC)
3. Plotting med ax.plot()

Vi holder fortsatt kapitalmengde konstant (f.eks. k=10k = 10k=10).

**🧠 Teoretisk oppsett**

Fra tidligere har vi:

Cs(x,k)=350x2k+500k+10000C\_s(x, k) = \frac{350x^2}{k} + 500k + 10000Cs​(x,k)=k350x2​+500k+10000

* **ATC:** Cs(x,k)x\frac{C\_s(x, k)}{x}xCs​(x,k)​
* **AVC:** 350x2/kx=350xk\frac{350x^2/k}{x} = \frac{350x}{k}x350x2/k​=k350x​
* **MC:** ddxCs(x,k)=700xk\frac{d}{dx}C\_s(x, k) = \frac{700x}{k}dxd​Cs​(x,k)=k700x​

**✅ Full løsning i kode (strukturert som i oppgave a)**

python

KopierRediger

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

# -----------------------------

# Symbolsk del – Funksjoner

# -----------------------------

x, k, n = sp.symbols('x k n')

def produksjon(n, k):

return sp.sqrt(n \* k)

def kostnad(n, k):

return 350 \* n + 500 \* k + 10000

# Løs n som funksjon av x og k

n\_expr = sp.solve(produksjon(n, k) - x, n)[0]

# Total kostnad

C\_expr = kostnad(n\_expr, k)

# Gjennomsnittskostnader og grensekostnad

ATC\_expr = sp.simplify(C\_expr / x)

AVC\_expr = sp.simplify((350 \* x) / k)

MC\_expr = sp.simplify(sp.diff(C\_expr, x))

# -----------------------------

# Lambdify til numeriske funksjoner

# -----------------------------

ATC\_fn = sp.lambdify((x, k), ATC\_expr, 'numpy')

AVC\_fn = sp.lambdify((x, k), AVC\_expr, 'numpy')

MC\_fn = sp.lambdify((x, k), MC\_expr, 'numpy')

# -----------------------------

# Plotting

# -----------------------------

x\_vals = np.linspace(0.1, 100, 400)

k\_val = 10

# Evaluér

ATC\_vals = ATC\_fn(x\_vals, k\_val)

AVC\_vals = AVC\_fn(x\_vals, k\_val)

MC\_vals = MC\_fn(x\_vals, k\_val)

# Plot med ax

fig, ax = plt.subplots(figsize=(10, 6))

ax.plot(x\_vals, ATC\_vals, label='totale gjennomsnittskostnad', color='blue')

ax.plot(x\_vals, AVC\_vals, label='variable gjennomsnittskostnad', color='orange')

ax.plot(x\_vals, MC\_vals, label='grensekostnad', color='green')

# Stil

ax.set\_xlabel('Produksjonsmengde (x)', loc='right')

ax.set\_ylabel('Kostnad per enhet', loc='top', rotation=0)

ax.set\_title('Gjennomsnittskostnad og grensekostnad på kort sikt (K=10)')

ax.legend()

ax.grid(True)

ax.spines['top'].set\_visible(False)

ax.spines['right'].set\_visible(False)

plt.tight\_layout()

plt.show()

**📌 Forklaring / Teori (kan brukes i oppgavebesvarelse):**

* **AVC** (oransje linje) øker jevnt med produksjon. Den er lineær: AVC(x)=350xk\text{AVC}(x) = \frac{350x}{k}AVC(x)=k350x​
* **ATC** (blå linje) har U-form – høye faste kostnader per enhet i starten → synker, men stiger igjen pga økt variabel kostnad
* **MC** (grønn linje) er lineær og krysser både **AVC** og **ATC** i deres minimumspunkt – klassisk økonomisk teori

Vil du at jeg skal formatere kommentarene slik at du kan lime det rett inn i en innlevering (del b)?

4o

**Du sa:**

Kunne du forklart grafen i oppgave b?

**ChatGPT sa:**

Ja, absolutt! Her er en **forklaring på grafen i oppgave b** – slik du kan bruke den direkte i en besvarelse eller som grunnlag for muntlig forklaring. Den tar utgangspunkt i figuren vi lagde med:

* Totale gjennomsnittskostnader (ATC)
* Variable gjennomsnittskostnader (AVC)
* Grensekostnader (MC)

Kapitalmengden er konstant: K=10K = 10K=10

**📊 Forklaring av grafen i oppgave b**

Grafen viser hvordan kostnader per produserte enhet utvikler seg med økende produksjonsmengde xxx, gitt en fast kapitalmengde. Vi ser på tre nøkkelfunksjoner:

**🔵 Totale gjennomsnittskostnader (ATC)**

ATC(x)=C(x)x=350xk+500k+10000x\text{ATC}(x) = \frac{C(x)}{x} = \frac{350x}{k} + \frac{500k + 10000}{x}ATC(x)=xC(x)​=k350x​+x500k+10000​

* ATC er summen av faste og variable kostnader per enhet.
* Kurven har en **U-form**:
  + Ved lav produksjon er faste kostnader (500k + 10000) fordelt på få enheter, så ATC er høy.
  + Når produksjonen øker, synker ATC fordi faste kostnader fordeles på flere enheter.
  + Etter et visst punkt stiger ATC igjen fordi variable kostnader (arbeidskraft) øker raskt.

**🟠 Variable gjennomsnittskostnader (AVC)**

AVC(x)=VC(x)x=350xk\text{AVC}(x) = \frac{VC(x)}{x} = \frac{350x}{k}AVC(x)=xVC(x)​=k350x​

* AVC stiger **lineært** med produksjonen.
* Dette er fordi arbeidskraftbehovet vokser i kvadrat med produksjon, gitt produksjonsfunksjonen x=n⋅kx = \sqrt{n} \cdot \sqrt{k}x=n​⋅k​.
* AVC tar **ikke med faste kostnader**.

**🟢 Grensekostnad (MC)**

MC(x)=dCdx=700xk\text{MC}(x) = \frac{dC}{dx} = \frac{700x}{k}MC(x)=dxdC​=k700x​

* MC viser hvor mye **ekstra kostnad** det er å produsere én enhet til.
* Denne øker **lineært**, fordi kostnaden ved ekstra produksjon krever mer arbeidskraft i økende tempo.
* Det er typisk i modeller med avtakende marginalprodukt av arbeidskraft.

**📌 Sammenhenger og økonomisk tolkning**

* **MC krysser både AVC og ATC i deres bunnpunkt**. Dette er en standardregel i mikroøkonomi:
  + Når MC < ATC: ATC synker
  + Når MC > ATC: ATC stiger
* Det samme gjelder forholdet mellom MC og AVC.
* Dette bekrefter at kostnadsstrukturene følger **økonomisk teori om kostnadskurver på kort sikt**.

**🧠 Konklusjon**

Grafen illustrerer hvordan produksjonskostnader per enhet påvirkes av faste og variable kostnader, og hvordan marginalkostnad bestemmer utviklingen i gjennomsnittskostnadene. Ved økende produksjon blir først produksjonen mer effektiv (ATC og AVC synker), men etter hvert tar variable kostnader over, og det blir dyrere å produsere mer.

Vil du ha dette som punktvis oppsummering, eller tilpasset som en akademisk tekst for innlevering?

4o

Øverst i skjemaet

O

Nederst i skjemaet