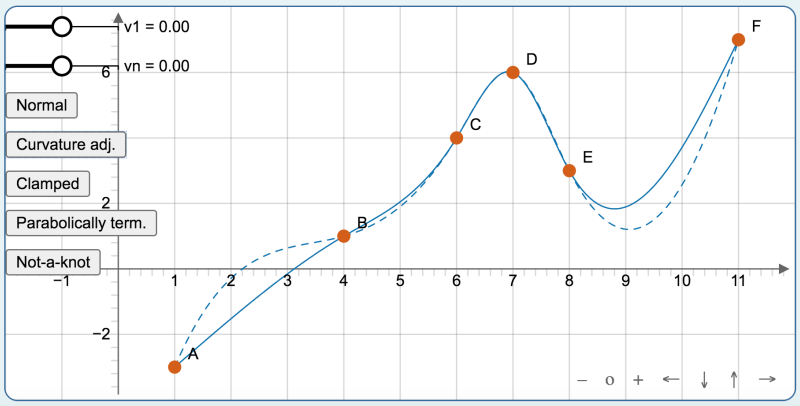
Løsning øving 5

Oppgave 1

Figuren under viser en kubisk spline mellom 6 punkter. Du skal velge riktige endebetingelser, og eventuelt stille på betingelsene for de deriverte og andrederiverte i start/endepunkt (For "clamped cubic spline" og "curvature adjusted cubic spline") slik at din spline matcher splinen vi har brukt til å lage den dottede grafen.



Denne spline-funksjonen kan tilpasses enten som "curvature adjustable spline" eller som "clamped cubic spline". Med disse to typene endepunktbetingelse kan man generere nøyaktig de samme spline-funksjonene. Forskjellen er kun at brukeren justerer krumningen i endepunktene i det første tilfellet og stigningstallet i det andre tilfellet, men om en spline kan genereres som "curvature adjustable spline" kan den også genereres som "clamped cubic spline". En "normal spline" har vendepunkt som endepunkt (der krumningen er null). Dette er ikke tilfellet her (som sikkert er litt vanskelig å se uten å velge den knappen). En "parabolically terminated spline" har andregrads funksjoner (parabler) mellom de første to og mellom de siste to punktene. Det er klart at det ikke er tilfellet her. "Not a knot"-betingelsen betyr at tredjegradsfunksjonene mellom de første to punktene og mellom de andre to punktene er identiske (og tilsvarende på slutten av spline-funksjonen). Det vil si at en enkel tredjegradsfunksjon går gjennom de første tre punktene, og tilsvarende for de siste tre punktene. Dette er ikke tilfellet her, men det er sikkert ikke så lett å se uten å trykke på knappen.

Oppgave 2

En normal kubisk spline $S(x)$ er gitt ved
$S_1(x) = -(x+1)^3 + A \cdot (x+1) - 5, \quad x \in [-1, 0]$
$S_2(x) = 3 \cdot x^3 + B \cdot x^2 - 5 \cdot x - 8, \quad x \in [0, 1]$
$S_3(x) = -2 \cdot (x-1) - 2 \cdot (x-1)^3 + 6 \cdot (x-1)^2 - 13, \quad x \in [1, 2]$
interpolerer punktene $(-1, -5)$, $(0, -8)$, $(1, -13)$ og $(2, -11)$.
Bestem verdien til parameterene A og B.

A =

B =

Her er $S_1(0) = A - 6$ og $S_2(0) = -8$, så $A = -2$.

Videre er $B - 10 = S_2(1) = S_3(1) = -13$, så $B = -3$. Legg merke til at også de første- og andrederiverte er like i disse punktene: Evaluert i x=0 gjelder $\frac{dS_1(x)}{dx} = \frac{dS_2(x)}{dx} = -5$ og $\frac{d^2S_1(x)}{dx^2} = \frac{d^2S_2(x)}{dx^2} = -6$. Tilsvarende gjelder i punkt x=1: $\frac{dS_2(x)}{dx} = \frac{dS_3(x)}{dx} = -2$ og $\frac{d^2S_2(x)}{dx^2} = \frac{d^2S_3(x)}{dx^2} = 12$.

Oppgave 3

En normal kubisk spline $S(x)$ er gitt ved
$S_1(x) = -(x+2)^3 + 2 \cdot (x+2) - 3, \quad x \in [-2, -1]$,
$S_2(x) = 3 \cdot (x+1)^3 + A \cdot (x+1)^2 - x - 3, \quad x \in [-1, 0]$ og
$S_3(x) = C \cdot x^3 + 6 \cdot x^2 + 2 \cdot x + B, \quad x \in [0, 1]$.
Den interpolerer punktene $(-2, -3)$, $(-1, -2)$, $(0, -3)$ og $(1, 3)$.
Bestem verdien til parameterene A, B og C.

A =

B =

C =

Her er $S_1(-1) = S_2(-1)$ og derfor er $-2 = -2$. Dette hjelper oss ikke.

Videre er $S_2(0) = S_3(0)$ og derfor er $A = B$. Siden det er gitt at spline-funksjonen går gjennom $(0, -3)$ er $A = B = -3$. Siden funksjonen går gjennom punkt $(1, 3)$ er $3 = S_3(1) = B + C + 8$. Siden $B = -3$ finner vi til slutt $C = -2$.

Oppgave 4

Vi har følgende liste med punkter

$$[(x_1, y_1), (x_2, y_2)], \dots] = [[1, -1], [2, 2], [3, 1], [4, 4]]$$

Bestem den normale kubiske splinen gjennom punktene:

$S_1 =$

$S_2 =$

$S_3 =$

Vi bruker ligningssett (videre "LS") 3.24 i Sauer. Koeffisientene som vi trenger er $\delta_1 = x_2 - x_1 = 2-1 = 1$, $\delta_2 = x_3 - x_2 = 3-2 = 1$, $\delta_3 = x_4 - x_3 = 4-3 = 1$ og $\Delta_1 = y_2 - y_1 = 2-(-1) = 3$, $\Delta_2 = y_3 - y_2 = 1-2 = -1$, $\Delta_3 = y_4 - y_3 = 4-1 = 3$. Formlene gir at LS-et for bestemmelse av den naturlige spline er $Ac = rhs$. Matrisen A blir:

$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$,
$rhs = [0, -12, 12, 0]^T$ med løsningen $\mathbf{c} = [0, -4, 4, 0]$.
Etter det finner vi fra 3.22:
$d_1 = \frac{c_2 - c_1}{3\delta_1} = \frac{-4-0}{3} = \frac{-4}{3}$
$d_2 = \frac{c_3 - c_2}{3\delta_2} = \frac{4-(-4)}{3} = \frac{8}{3}$
$d_3 = \frac{c_4 - c_3}{3\delta_3} = \frac{0-4}{3} = \frac{-4}{3}$ og fra 3.23:
$b_1 = \frac{\Delta_1}{\delta_1} - \frac{\delta_1}{3}(2c_1 + c_2) = \frac{3}{1} - \frac{1}{3}(2 \cdot 0 + -4) = \frac{13}{3}$
$b_2 = \frac{\Delta_2}{\delta_2} - \frac{\delta_2}{3}(2c_2 + c_3) = \frac{-1}{1} - \frac{1}{3}(2 \cdot (-4) + 4) = \frac{1}{3}$
$b_3 = \frac{\Delta_3}{\delta_3} - \frac{\delta_3}{3}(2c_3 + c_4) = \frac{3}{1} - \frac{1}{3}(2 \cdot (4) + 0) = \frac{1}{3}$
Spline-funksjonen blir dermed:
$S_1(x) = -1 + \frac{13}{3}(x-1) + 0(x-1)^2 - \frac{4}{3}(x-1)^3$
$S_2(x) = 2 + \frac{1}{3}(x-2) - 4(x-2)^2 + \frac{8}{3}(x-2)^3$
$S_3(x) = 1 + \frac{1}{3}(x-3) + 4(x-3)^2 - \frac{4}{3}(x-3)^3$

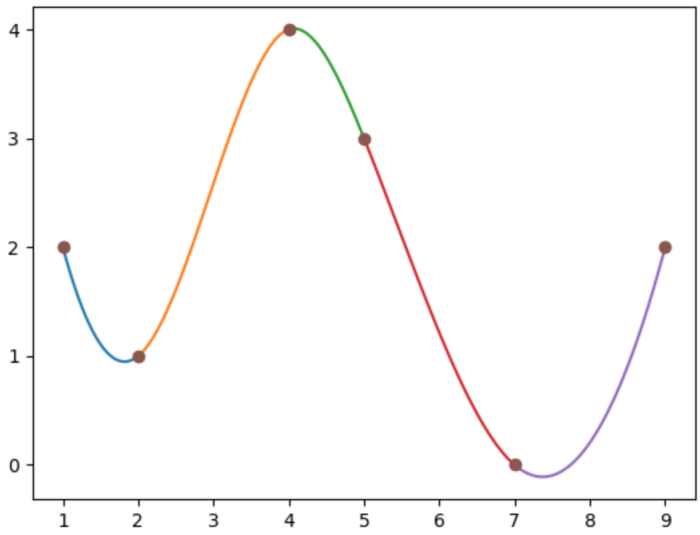
Legg merke til at en i forelesningene bruker b som både rhs i LS-et og b som koeffisient for $x - x_i$ i $S_i(x)$.

Oppgave 5

Fullfør funksjonen som setter opp det linjære likningssystemet for å
løse et splines-problem med "not-a-knot" endepunktsbetingelser

Funksjonen skal ta en liste med x- og y-koordinater som tilsvarer punkter x_i, y_i som vi skal interpolere med kubiske splines.
Videre skal funksjonen lage og returnere en matrise A og vektor (array) b som tilsvarer det linjære likningssystemet (3.24 i Sauer), med riktige endepunktsbetingelser (not-a-knot)

Dersom dere kjører koden i egen IDE og kommenterer ut plottekoden burde dere få plottet under



Løsning:

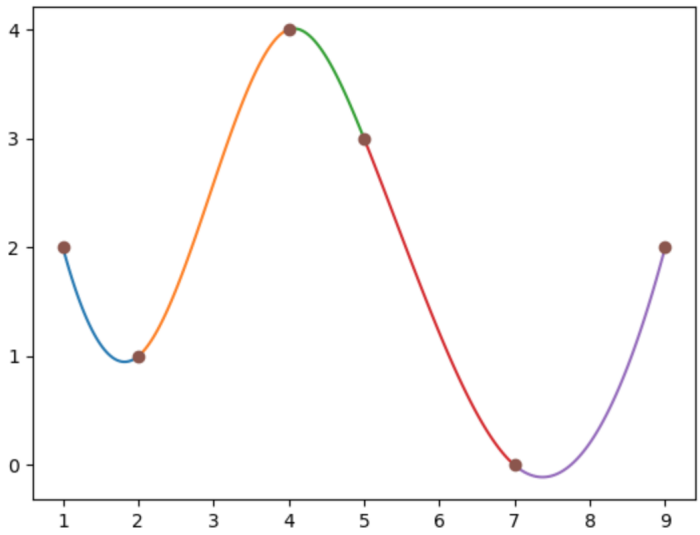
```python
import numpy as np

def getNAK_System(x_coords, y_coords):
    n = x_coords.size
    delta = np.array([x_coords[i]-x_coords[i-1] for i in range(1,n)])
    BigDelta = np.array([y_coords[i]-y_coords[i-1] for i in range(1,n)])
    A = np.zeros((n,n))

    mainDiag = np.ones(n)
    mainDiag[-1] = -1
    mainDiag[1:-1] = np.array([2*delta[i]+2*delta[i+1] for i in range(n-2) ])

    upperDiag = np.array([delta[i] for i in range(n-1)])
    upperDiag[0]=-1
    lowerDiag = np.array([delta[i] for i in range(n-1)])
    lowerDiag[-1] = 1

    np.fill_diagonal(A,mainDiag)
    np.fill_diagonal(A[:-1,1:],upperDiag)
    np.fill_diagonal(A[1:,:-1],lowerDiag)

    b = np.zeros(n)
    b[1:-1]=np.array(
        [3*(BigDelta[i]/delta[i]-BigDelta[i-1]/delta[i-1]) for i in range(1,n-1)]
    )

    return A, b

x = np.array([1,2,4,5.,7,9])
y = np.array([2,1,4,3.,0,2])
A, bs = getNAK_System(x,y)
```

Oppgave 6

Vi skal fortsette med programmet fra forrige oppgave. Nå vil vi utvide det med en funksjon `getNAK_splines(A,b,x_coords,y_coords)`

Funksjonen skal ta systemetmatrise og vekter A, b fra forrige oppgave, løse likningssettet med numpy.linalg.solve, og returnere fire lister a,b,c,d som inneholder koeffisientene i de $n-1$ kubiske splinesene med not-a-knot endepunktbetingelser.
(Se 3.22 og 3.23 i Sauer)

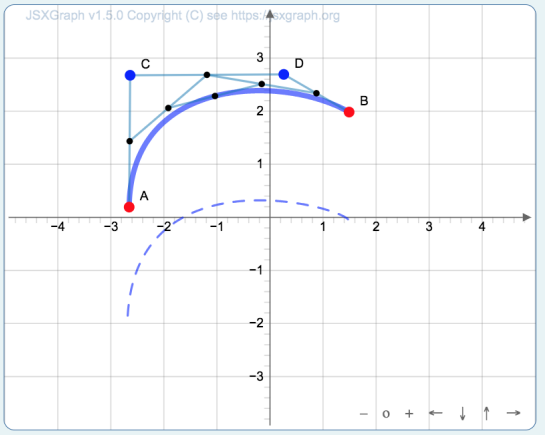
Dersom dere kjører koden i egen IDE og kommenterer inn plottekoden burde dere få figuren under



Løsning:

```python
import numpy as np
from numpy.linalg import solve

def getNAK_System(x_coords, y_coords):
    n = x_coords.size
    delta = np.array([x_coords[i]-x_coords[i-1] for i in range(1,n)])
    BigDelta = np.array([y_coords[i]-y_coords[i-1] for i in range(1,n)])
    A = np.zeros((n,n))

    mainDiag = np.ones(n)
    mainDiag[-1] = -1
    mainDiag[1:-1] = np.array([2*delta[i]+2*delta[i+1] for i in range(n-2) ])

    upperDiag = np.array([delta[i] for i in range(n-1)])
    upperDiag[0]=-1
    lowerDiag = np.array([delta[i] for i in range(n-1)])
    lowerDiag[-1] = 1

    np.fill_diagonal(A,mainDiag)
    np.fill_diagonal(A[:-1,1:],upperDiag)
    np.fill_diagonal(A[1:,:-1],lowerDiag)

    b = np.zeros(n)
    b[1:-1]=np.array(
        [3*(BigDelta[i]/delta[i]-BigDelta[i-1]/delta[i-1]) for i in range(1,n-1)]
    )

    return A, b

def getNAK_splines(A,b, x_coords, y_coords):
    n = x_coords.size
    delta = np.array([x_coords[i]-x_coords[i-1] for i in range(1,n)])
    BigDelta = np.array([y_coords[i]-y_coords[i-1] for i in range(1,n)])
    c = solve(A,b)
    d = np.array([(c[i+1]-c[i])/(3*delta[i]) for i in range(n-1)])
    b = np.array(
        [BigDelta[i]/delta[i]-delta[i]*(2*c[i]+c[i+1])/3 for i in range(n-1)]
    )
    a = y_coords
    return a,b,c,d



x = np.array([1,2,4,5.,7,9])
y = np.array([2,1,4,3.,0,2])
n = x.size
A, bs = getNAK_System(x,y)
a,b,c,d = getNAK_splines(A,bs,x,y)
```

Oppgave 7



Løsningen er vist men translert oppover med forflytning (0, 2) slik at kurven fra oppgaven er synlig. Endepunktene A og B er endepunktene til bezierkurven mens punkt C og D er kontrollpunktene. Kontrollpunkt C definerer bl.a. retningen til tangenten i punkt A mens kontrollpunkt D definerer bl.a. retningen til tangenten i punkt B.

Oppgave 8

Finn den kubiske Bezierkurven $x(t)$, $y(t)$ som beskrives av endepunktene

$$P_1 = [-2, 2]$$
$$P_2 = [2, 2]$$

og kontrollpunktene:

$$P_3 = [2, -1]$$
$$P_4 = [-2, -2]$$

$x(t) =$ _____

$y(t) =$ _____

Merk: for å bruke de oppgitte formlene må vi definere punktene som
$\mathbf{x}_1 = P_1 = [-2, 2]$ og $\mathbf{x}_4 = P_2 = [2, 2]$
(som er de to endepunktene), og
$\mathbf{x}_2 = P_3 = [2, -1]$ og $\mathbf{x}_3 = P_4 = [-2, -2]$
(som er de to kontrollpunktene der rekkefølgen strengt tatt ikke er opplyst i oppgaven, d.v.s. det kanskje er mulig å tolke oppgaven som om $\mathbf{x}_2 = P_4 = [-2, -2]$ og $\mathbf{x}_3 = P_3 = [2, -1]$ men dette er vel ulogisk). Vi følger videre definisjonen i kapittel 3.5 i Sauer:

$b_x = 3(x_2 - x_1) = 3(2 - (-2))) = 12$
$c_x = 3(x_3 - x_2) - b_x = 3(-2 - 2) - 12 = -24$
$d_x = x_4 - x_1 - b_x - c_x = 2 - (-2) - 12 - (-24) = 16$
$b_y = 3(y_2 - y_1) = 3((-1) - 2) = -9$
$c_y = 3(y_3 - y_2) - b_y = 3(-2 - (-1)) - (-9) = 6$
$d_y = y_4 - y_1 - b_y - c_y = 2 - 2 - (-9) - 6 = 3$

$x(t) = x_1 + b_x t + c_x t^2 + d_x t^3 = -2 + 12t - 24t^2 + 16t^3$
$y(t) = y_1 + b_y t + c_y t^2 + d_y t^3 = 2 - 9t + 6t^2 + 3t^3$

Oppgave 9

Finn første endepunkt P_1, de to kontrollpunktene P_2, P_3 og siste endepunkt P_4
Til bezierkurven beskrevet av:

$$x(t) = 9 \cdot t^3 - 9 \cdot t^2 + 3 \cdot t - 2$$
$$y(t) = 8 \cdot t^3 - 9 \cdot t^2 + 3 \cdot t$$

$P_1 =$
$P_2 =$
$P_3 =$
$P_4 =$

Gitt $x(t) = 9t^3 - 9t^2 + 3t - 2$ og $y(t) = 8t^3 - 9t^2 + 3t - 0$.

Først gir $t = 0$ at $P_1 = (-2, 0)$.

Videre gir $t = 1$ at $P_4 = (1, 2)$.

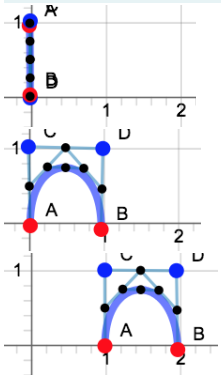
Av $x'(t) = 27t^2 - 18t + 3$ og $y'(t) = 24t^2 - 18t + 3$ får vi at $P'(0) = (3, 3)$ og $P'(1) = (12, 9)$.

Av formlene $3(P_2 - P_1) = P'(0)$ og $3(P_4 - P_3) = P'(1)$ fås da $3(P_2 - (-2, 0))) = (3, 3)$ og $3((1, 2)) - P_3) = (12, 9)$ slik at $P_2 = (-1, 1)$ og $P_3 = (-3, -1)$.

Oppgave 10

Finn bokstaven som er gitt ved bezierkurvene med punkter:

$$
\begin{aligned}
\boldsymbol{B}_1 &= [[0,1],[0,1],[0,0],[0,0]] \\
\boldsymbol{B}_2 &= [[0,0],[0,1],[1,1],[1,0]] \\
\boldsymbol{B}_3 &= [[1,0],[1,1],[2,1],[2,0]]
\end{aligned}
$$



Vi plotter de tre bezierkurvene og ser at de tre kurvene til sammen former bokstaven m (om vi plotter alle på samme graf).