Inna Williams
###

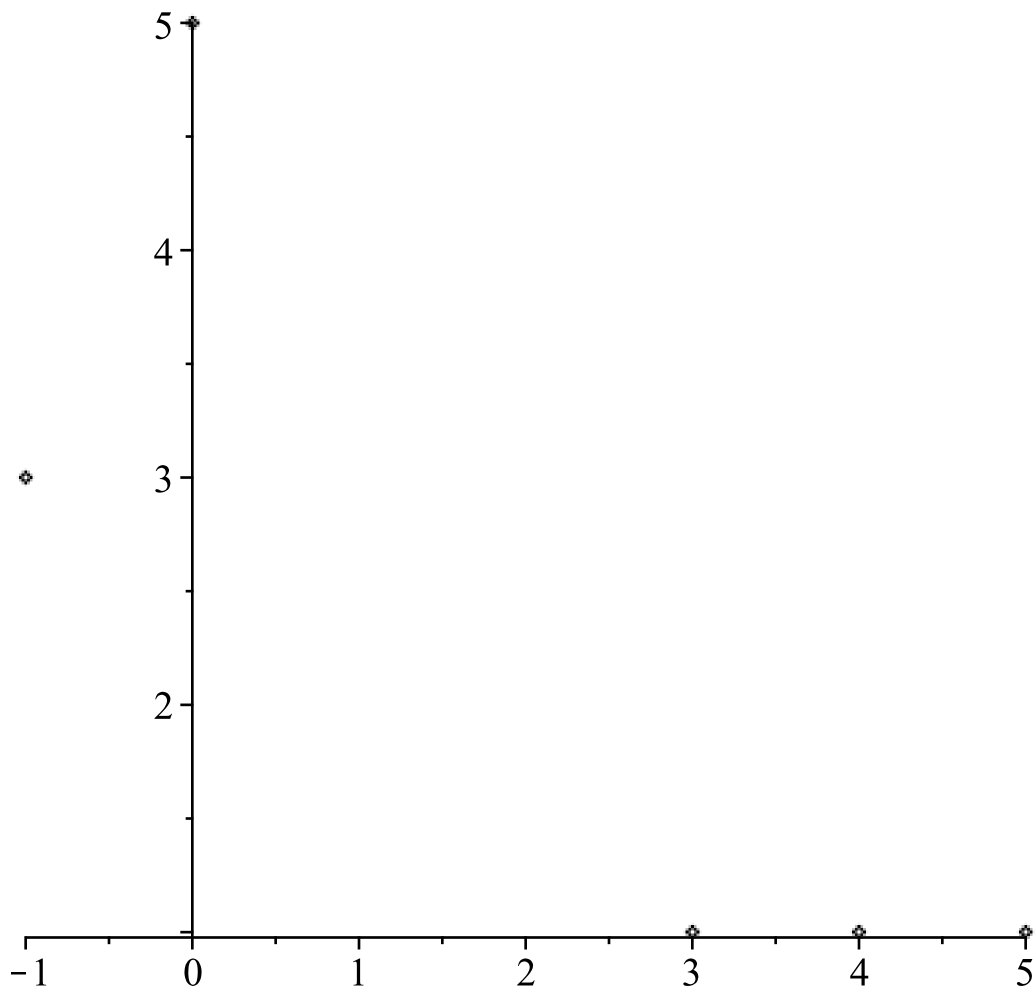
Section 3.4
###

1b
1. Find the equations and plot the natural cubic spline that interpolates the data points
(b) (-1,3), (0,5), (3.0,1.0), (4,1.0), (5,1.0).
****

```
> with(plots):with(CurveFitting);
```

[ArrayInterpolation, BSpline, BSplineCurve, Interactive, LeastSquares, Lowess, PolynomialInterpolation, RationalInterpolation, Spline, ThieleInterpolation] **(1)**

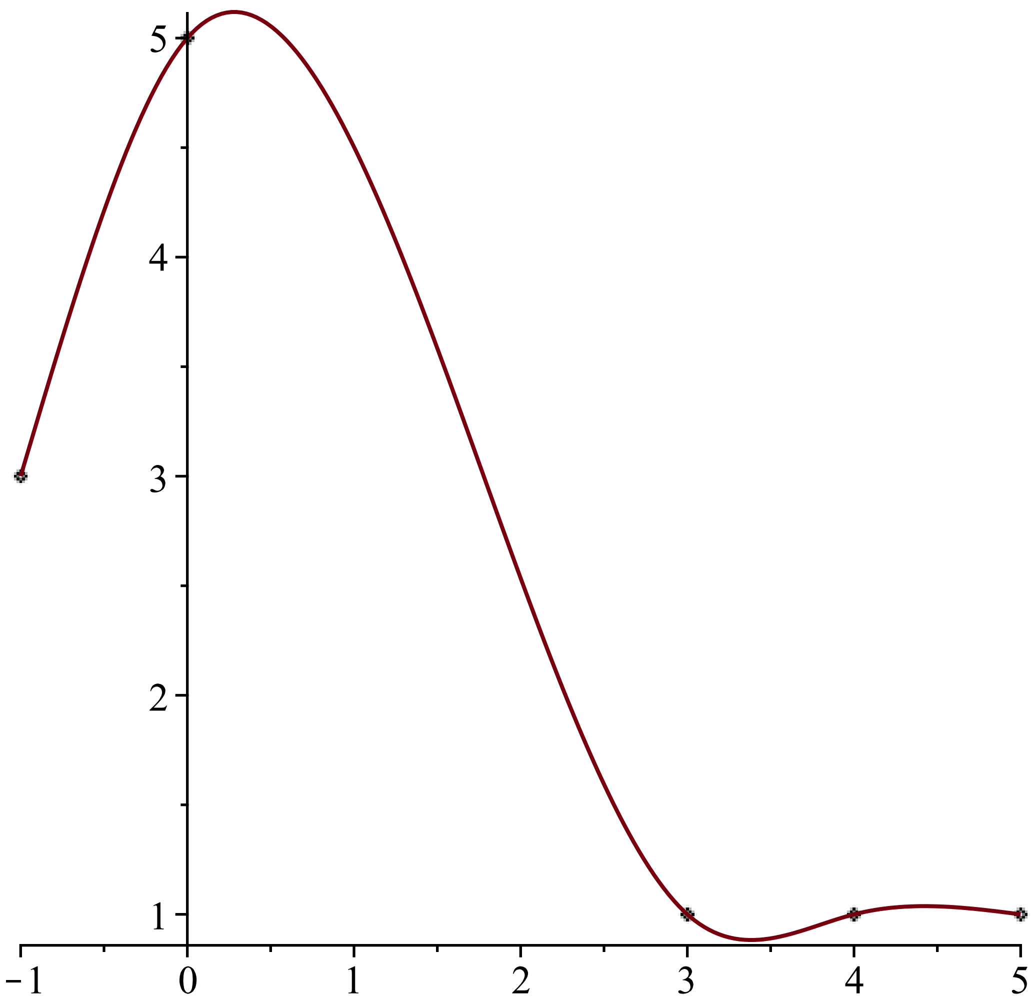
```
> data1 := [[-1, 3], [0, 5], [3.0, 1.0], [4, 1], [5, 1]]:p1 := plot(data1, style
    =point, color=black):display(p1)
```




```
> spline1 := evalf(Spline(data1, x, endpoints='natural'))
```

$$spline1 := \begin{cases} 5.56289308176101 + 2.56289308176101\ x - 0.562893081761006\ (x+1.)^3 & x < 0. \\ 5. + 0.874213836477987\ x - 1.68867924528302\ x^2 + 0.317610062893082\ x^3 & x < 3.0 \\ 3.04716981132075 - 0.682389937106918\ x + 1.16981132075472\ (x-3.0)^2 - 0.487421383647799\ (x-3.0)^3 & x < 4. \\ 0.220125786163522 + 0.194968553459119\ x - 0.292452830188679\ (x-4.)^2 + 0.0974842767295597\ (x-4.)^3 & otherwise \end{cases}$$

```
> p2 := plot(spline1, x = -1 ..5) : display(p1, p2)
```

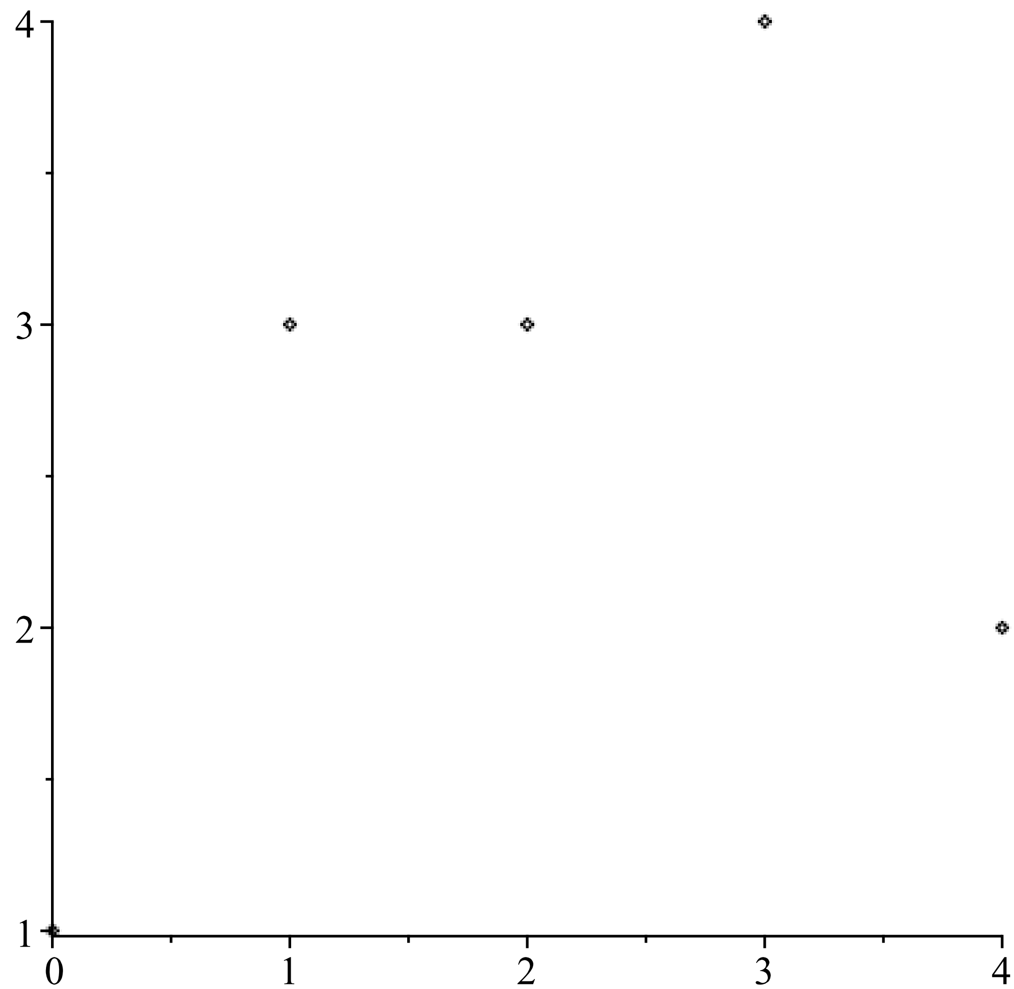


```
>
```

##

5. Find and plot the cubic spline S satisfying
S(0) = 1,   S(1) = 3, S(2) = 3, S(3) = 4, S(4) = 2
and with S'(0) = 0 and S'(4) = 1.

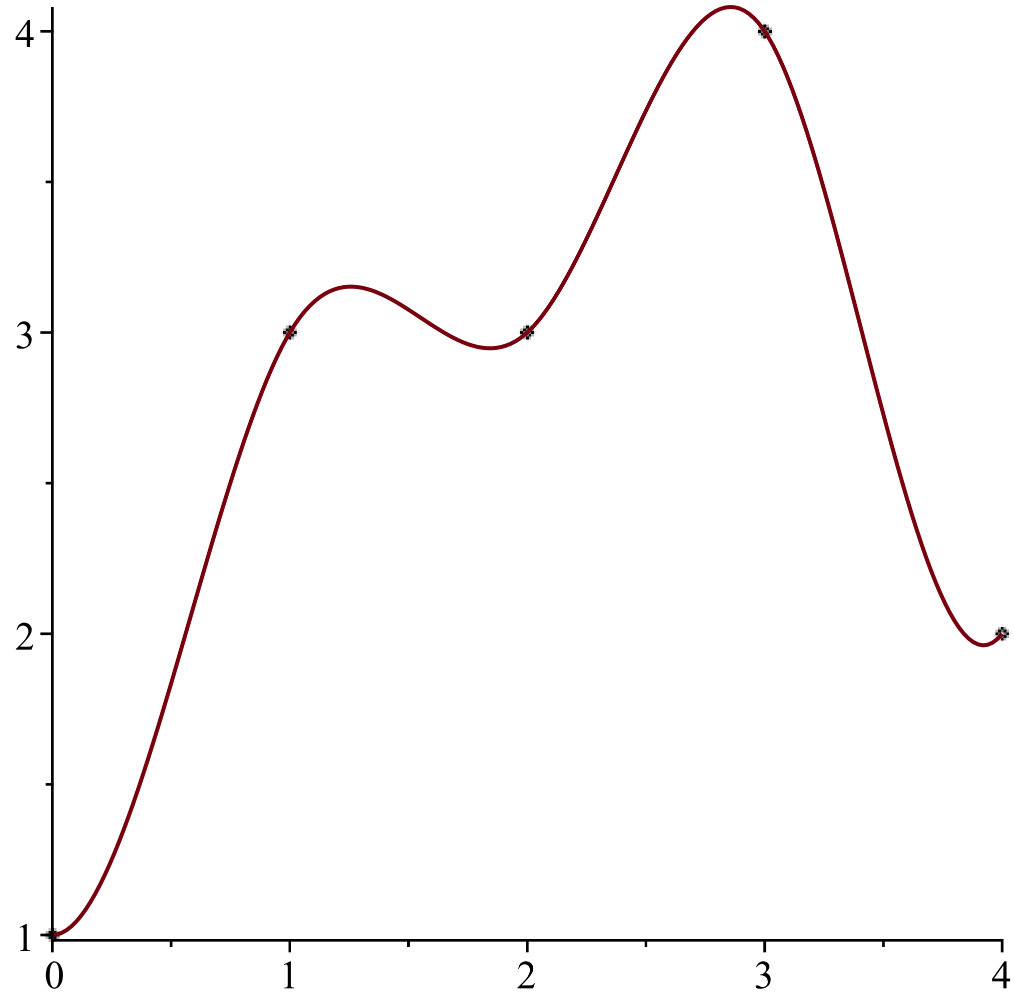
```
> data1 := [[0, 1], [1, 3], [2, 3], [3, 4], [4, 2]] : p1 := plot(data1, style = point,
      color = black) : display(p1)
```




```
> clamped_spline := evalf(Spline(data1, x, endpoints = [0, 1]))
```

$$clamped\_spline := \begin{cases} -2.678571429\ x^3 + 4.678571429\ x^2 + 1 & x < 1. \\ 2.035714286\ x^3 - 9.464285714\ x^2 + 14.14285714\ x - 3.714285714 & x < 2. \\ -2.464285714\ x^3 + 17.53571429\ x^2 - 39.85714286\ x + 32.28571429 & x < 3. \\ 3.821428571\ x^3 - 39.03571429\ x^2 + 129.8571429\ x - 137.4285714 & otherwise \end{cases}$$

```
> p2 := plot(clamped_spline, x = 0..4) : display(p1, p2)
```



##

7. Find the clamped cubic spline that interpolates f (x) = cosx at five evenly spaced points in [0,π/2], including the endpoints. What is the best choice for S'(0) and S'(π/2) to minimize interpolation error? Plot the spline and cosx on [0,2].

f'(x)=(cos(x))' = -sinx

S'(0) = -sin(0) = 0

$S'(\frac{\pi}{2}) = -\sin\left(\frac{\pi}{2}\right) \ = \ -1$

$$\frac{\pi}{2}$$

$$-1 = -1 \tag{2}$$

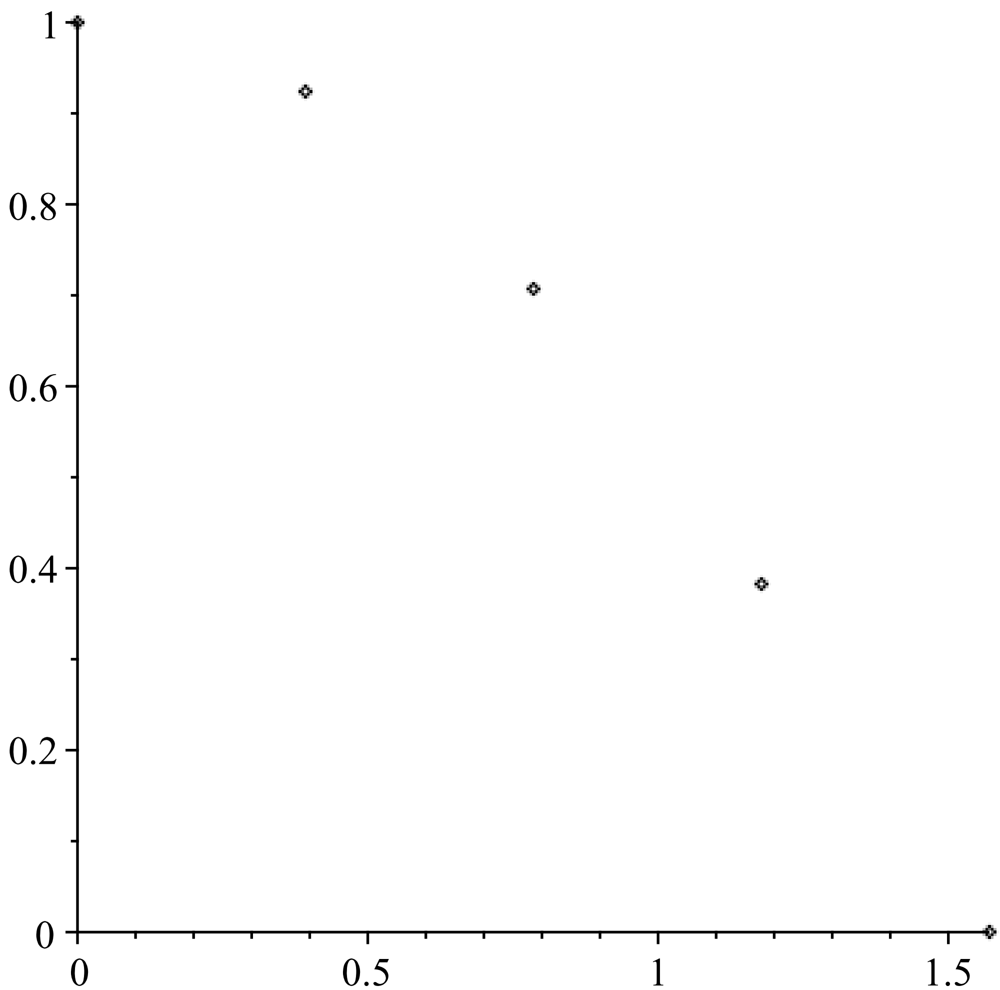
```
> y := z→cos(z)
```

$$y := z \mapsto \cos(z) \tag{3}$$

$> \ z := Array\left(\left[0, \frac{\pi}{8}, \ 2 \cdot \frac{\pi}{8}, \ 3 \cdot \frac{\pi}{8}, \ \frac{\pi}{2}\right]\right)$

$$z := \left[\begin{array}{ccccc} 0 & \frac{\pi}{8} & \frac{\pi}{4} & \frac{3\ \pi}{8} & \frac{\pi}{2} \end{array}\right] \quad \textbf{(4)}$$

```
> points := ([[z(1), y(z(1))], [z(2), y(z(2))], [z(3), y(z(3))], [z(4),
      y(z(4))], [z(5), y(z(5))]]) : p1 := plot(points, style = point, color
      = black) : display(p1)
```




```
> points := evalf(points)
```

$$points := [[0., 1.], [0.3926990818, 0.9238795325], [0.7853981635, 0.7071067810], [1.178097245, 0.3826834325], [1.570796327, 0.]] \quad \textbf{(5)}$$

```
> slope_left := 0
```

$$slope\_left := 0 \quad \textbf{(6)}$$

```
> slope_right := -1
```

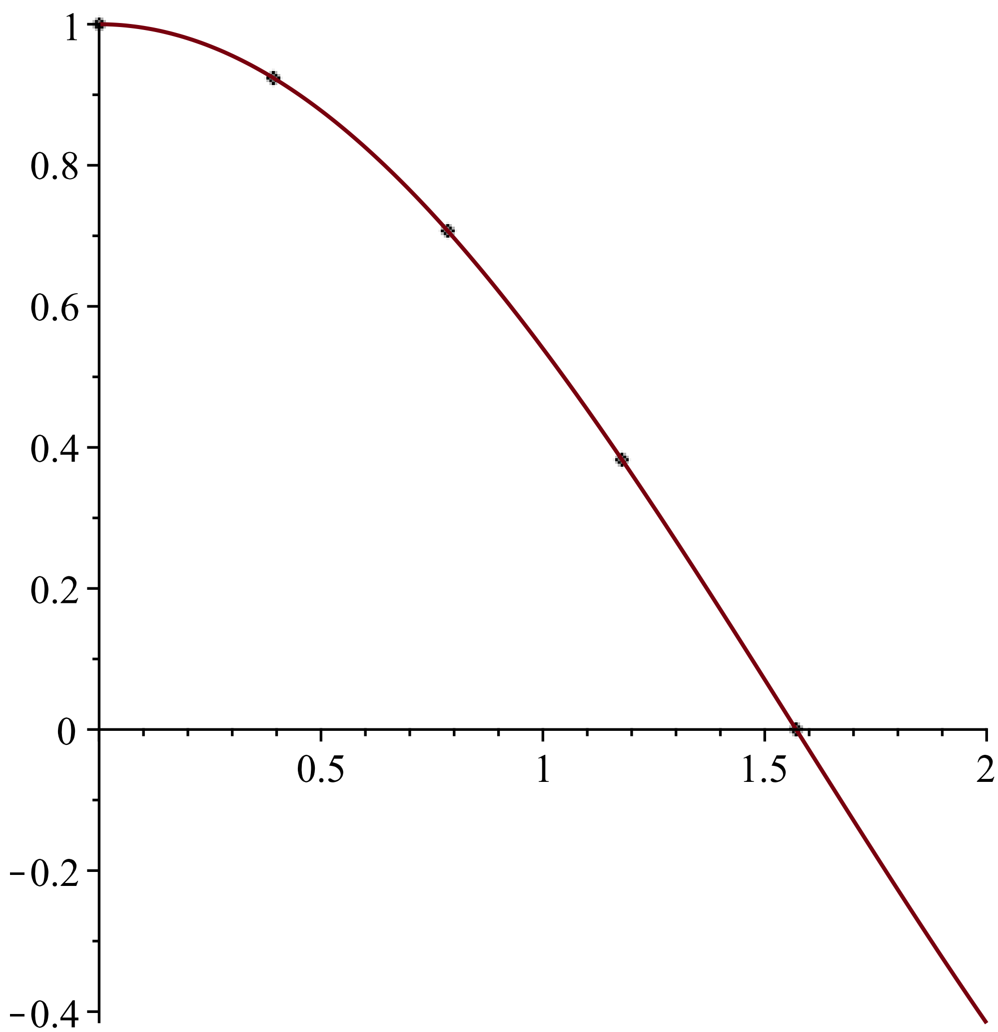
$$slope\_right := -1 \quad \textbf{(7)}$$

```
> clamedspline := Spline(points, x, endpoints = [slope_left, slope_right])
```

$$clamedspline := \begin{cases} 1 - 2.7756\,10^{-17}\ \ x - 0.5064\,x^2 + 0.0327\ \ x^3 & x < 0.3927 \\ 1.0741 - 0.3826\,x - 0.4679\,(x - 0.3927)^2 + 0.0931\ \ (x - 0.3927)^3 & x < 0.7854 \\ 1.2624 - 0.7070\,x - 0.3582\,(x - 0.7854)^2 + 0.1396\ \ (x - 0.7854)^3 & x < 1.1781 \\ 1.4709 - 0.9237\,x - 0.1937\,(x - 1.1781)^2 + 0.1639(x - 1.1781)^3 & otherwise \end{cases}$$

>

```
> p2 := plot(clamedspline, x = 0..2) : display(p1, p2)
```



>

>

##

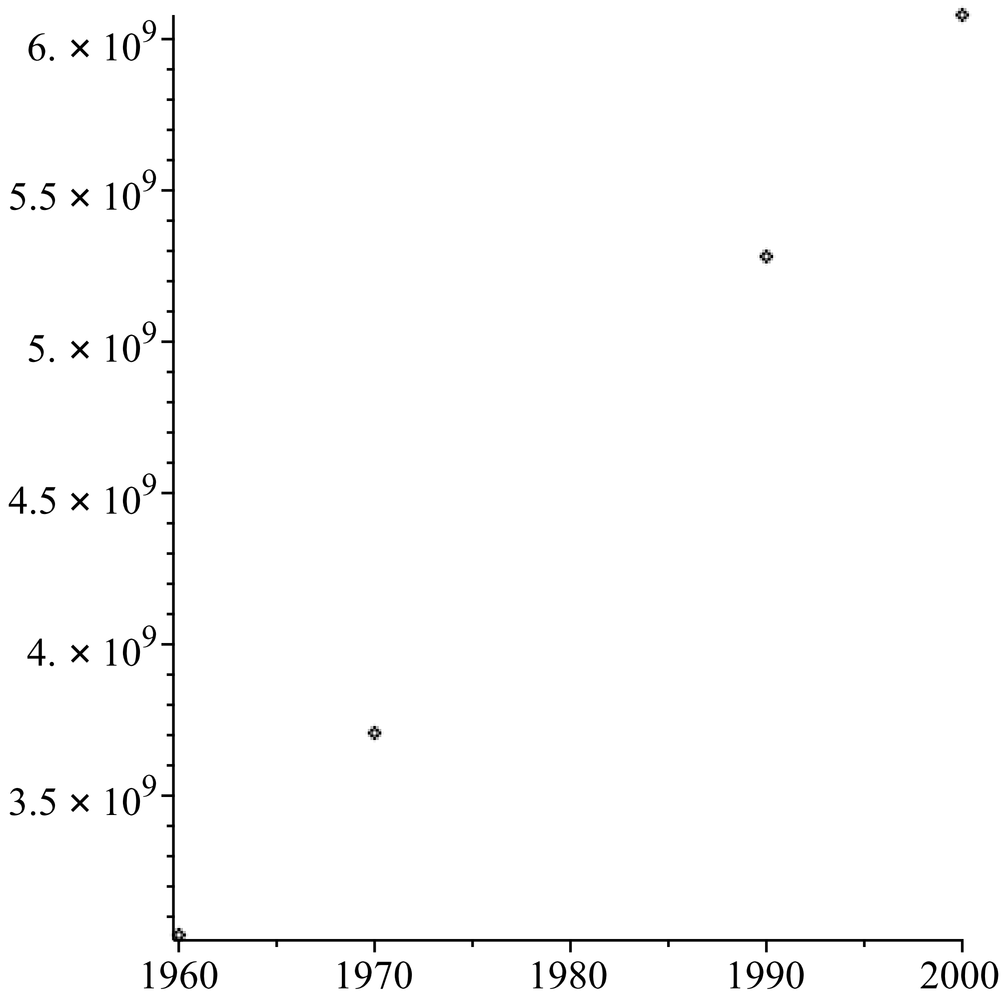
11. (a) Consider the natural cubic spline through the world population data points in Computer Problem 3.1.1. Evaluate the year 1980 and compare with the correct population. Compare with the 1980 estimate of 4452584592

year population
1960 3039585530
1970 3707475887
1990 5281653820
2000 6079603571

> *population_1980* := 4452584592

$$population\_1980 := 4452584592 \quad \textbf{(8)}$$

```
> data1 := [[1960, 3039585530], [1970, 3707475887], [1990, 5281653820],
    [2000, 6079603571]] : p1 := plot(data1, style = point, color = black) :
    display(p1)
```




```
> spline_natural := evalf(Spline(data1, x, endpoints='natural'))
```

$$spline\_natural := \begin{cases} 21670.94025\ x^3 - 1.274251287\ 10^8\ x^2 + 2.498178741\ 10^{11}\ x - 1.632957442\ 10^{14} & x < 1970. \\ -13542.22812\ x^3 + 8.068469643\ 10^7\ x^2 - 1.601584813\ 10^{11}\ x + 1.059220626\ 10^{14} & x < 1990. \\ 5413.516000\ x^3 - 3.2481096\ 10^7\ x^2 + 6.504144562\ 10^{10}\ x - 4.346055564\ 10^{13} & otherwise \end{cases}$$

> computed_1980_natural := eval(spline_natural, x = 1980)

$$computed\_1980\_natural := 4.4703\ 10^9 \quad \textbf{(9)}$$

```
> absolute_error := abs(population_1980 - computed_1980_natural)
```

$$absolute\_error := 1.7715408\ 10^7 \quad \textbf{(10)}$$

```
> relative_error := absolute_error/population_1980
```

$$relative\_error := 0.003978679716 \quad \textbf{(11)}$$

Answer:

computed_1980_natural = 4470300000
absolute_error = 17715408
relative_error ~ 0.40%

##

11. (b) Using a linear spline, estimate the slopes at 1960 and 2000, and use these slopes to find the clamped cubic spline through the data. Plot the spline and estimate the 1980 population. Which estimates better, natural or clamped?

```
> d := Array(data1)
```

$$d := \begin{bmatrix} 1960 & 3039585530 \\ 1970 & 3707475887 \\ 1990 & 5281653820 \\ 2000 & 6079603571 \end{bmatrix} \quad (12)$$

```
>
```

$$> \; slope\_1960 := evalf\left(\frac{d[2][2] - d[1][2]}{d[2][1] - d[1][1]}\right)$$

$$slope\_1960 := 6.678903570\ 10^{7} \quad (13)$$

$$> \; slope\_2000 := evalf\left(\frac{d[4][2] - d[3][2]}{d[4][1] - d[3][1]}\right)$$

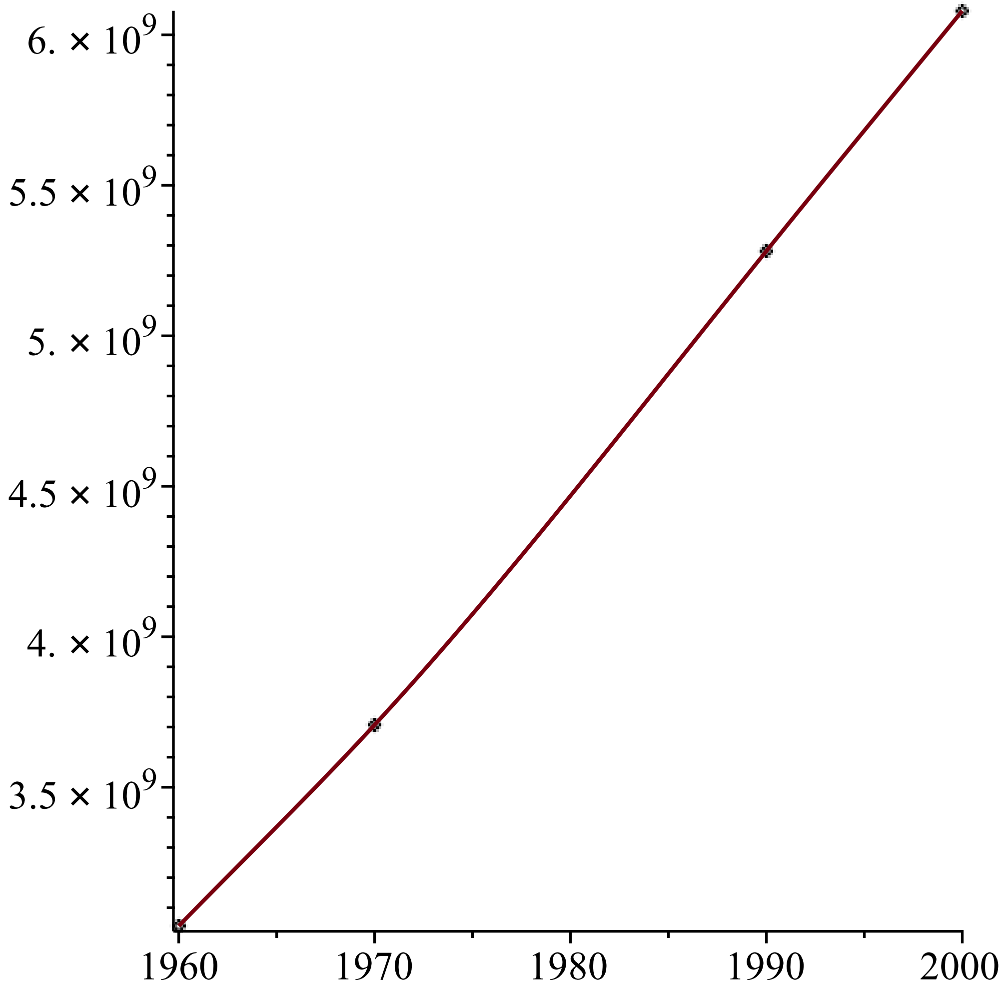
$$slope\_2000 := 7.979497510\ 10^{7} \quad (14)$$

```
> clamedspline := Spline(data1, x, endpoints = [slope_1960, slope_2000])
```

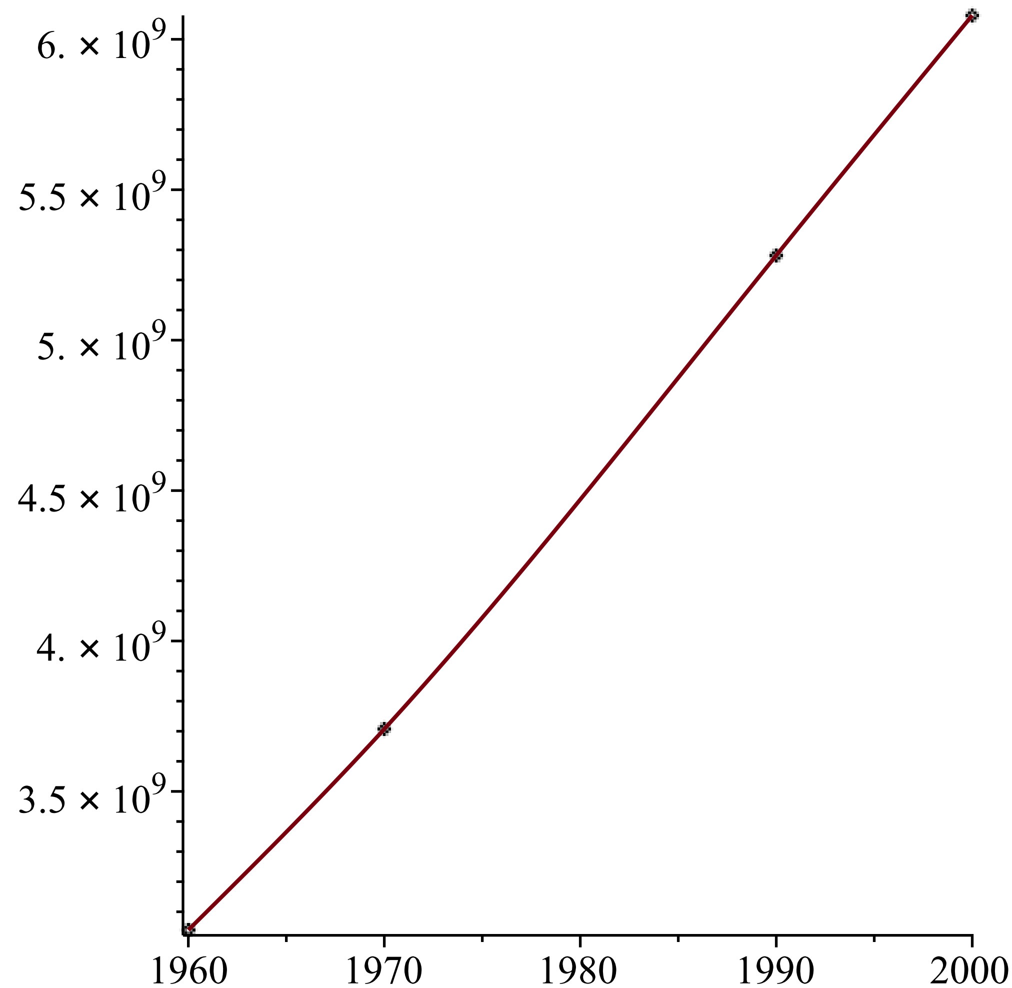
clamedspline :=

$$\begin{cases} 35924.78297\ x^3 - 2.115942182\ 10^{8}\ x^2 + 4.154901855\ 10^{11}\ x - 2.719943224\ 10^{14} & x < 1970 \\ -15389.07005\ x^3 + 9.167065310\ 10^{7}\ x^2 - 1.819416105\ 10^{11}\ x + 1.203192232\ 10^{14} & x < 1990 \\ 10056.59631\ x^3 - 6.023997512\ 10^{7}\ x^2 + 1.203605397\ 10^{11}\ x - 8.020786972\ 10^{13} & otherwise \end{cases}$$

```
> p2 := plot(clamedspline, x = 1960..2000) : display(p1, p2)
```




```
> p2 := plot(spline_natural, x = 1960..2000) : display(p1, p2)
```



\>

\> computed_1980_clamed := eval(clamedspline, x = 1980)

$$computed\_1980\_clamed := 4.4686\ 10^{9} \quad \textbf{(15)}$$

```
> absolute_error := abs(population_1980 - computed_1980_clamed)
```

$$absolute\_error := 1.6015408\ 10^{7} \quad \textbf{(16)}$$

```
> relative_error := absolute_error/population_1980
```

$$relative\_error := 0.003596878997 \quad \textbf{(17)}$$

Answer:
computed_1980_clamed = 4468600000
absolute_error_clamed = 16015408 compare to 17715408 for natural spline
relative_errorclamed ~ 0.36% compare to 0.40% for natural spline

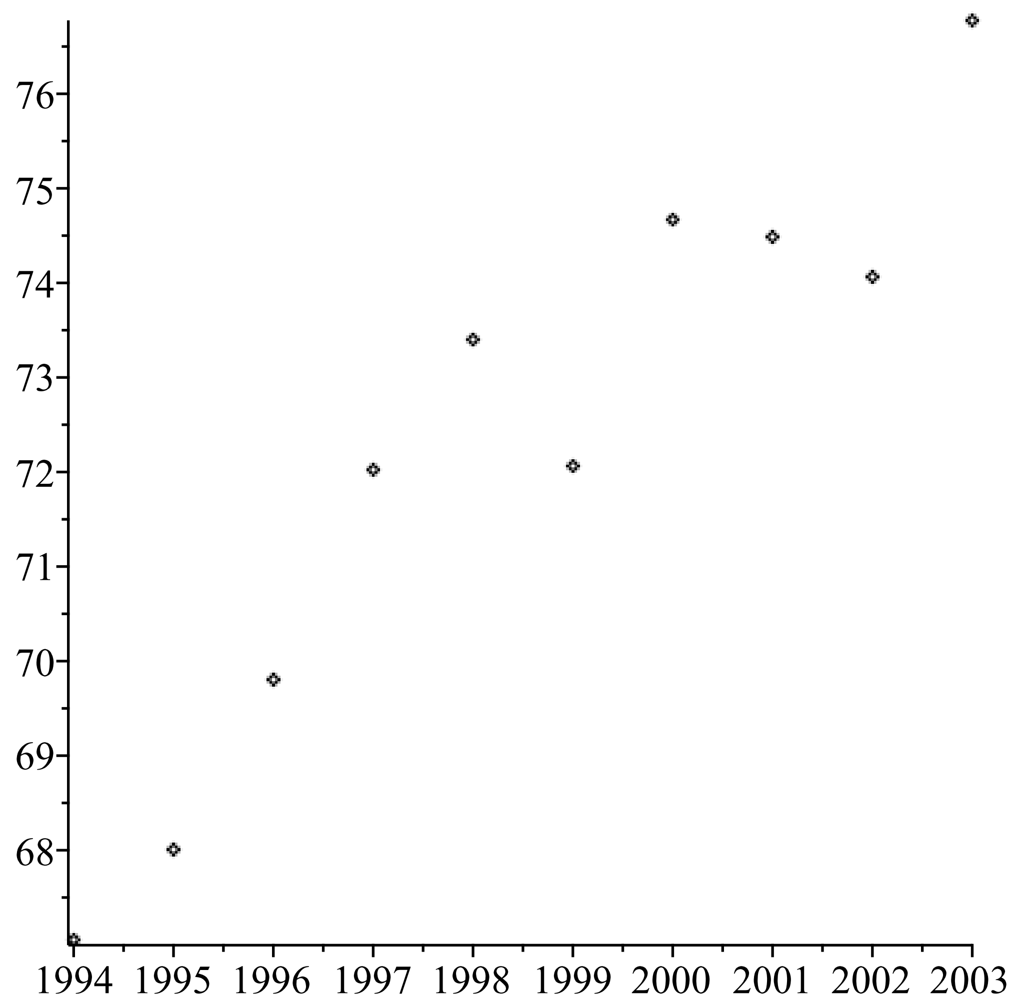
Clamed is better because absolute and relative errors are smaller then the ones for natural spline
##

Section 3.2

##

3. The total world oil production in millions of barrels per day is shown in the table that follows. Determine and plot the degree 9 polynomial through the data. Use it to estimate 2010 oil production. Does the Runge phenomenon occur in this example? In your opinion, is the interpolating polynomial a good model of the data? Explain.
year bbl/day (×106)
1994 67.052
1995 68.008
1996 69.803
1997 72.024
1998 73.400
1999 72.063
2000 74.669
2001 74.487
2002 74.065
2003 76.777

```
> interpolated_poly_degree10 := [[1994, 67.052], [1995, 68.008], [1996,
    69.803], [1997, 72.024], [1998, 73.400], [1999, 72.063], [2000, 74.669],
    [2001, 74.487], [2002, 74.065], [2003, 76.777]]:
  p3 := plot(interpolated_poly_degree10, style = point, color = black) :
    display(p3);
```




```
> p := x→PolynomialInterpolation(interpolated_poly_degree10, x)
```

$$p := x \mapsto PolynomialInterpolation(interpolated\_poly\_degree10, x) \quad \textbf{(18)}$$

```
> poly := PolynomialInterpolation(interpolated_poly_degree10, x)
```

$$poly := -0.0007352458096\ x^9 + 13.22405475\ x^8 - 105709.5140\ x^7 + 4.929242469\ 10^8\ x^6 - 1.477612769\ 10^{12}\ x^5 + 2.952906560\ 10^{15}\ x^4 - 3.934118065\ 10^{18}\ x^3 + 3.369453149\ 10^{21}\ x^2 - 1.683403103\ 10^{24}\ x + 3.737955965\ 10^{26} \quad \textbf{(19)}$$

> $check\_poly := [p(1994), p(1995), p(1996), p(1997), p(1998), p(1999), p(2000), p(2001), p(2002), p(2003), p(2010)]$

$$check\_poly := [67.052, 68.008, 69.80300000, 72.02400000, 73.40000000, 72.06300000, 74.66900000, 74.48700003, 74.06499989, 76.77699988, -1.951646127\ 10^6] \quad \textbf{(20)}$$

```
> p(2010)
```

$$-1.951646127\ 10^6 \quad \textbf{(21)}$$

We can see that the data from p(1994)-p(2003) are correct but the p(2010)
is not consistent with the data. This is due to the example of the Runge phenomenon
I can conclude that is does not make degree-9 interpolation polynomial being usable model
for extrapolating data.