Model empirik

https://github.com/dudung/sk5003-02-2022-2

Sparisoma Viridi
Master Program in Computational Science, Nuclear Physics and Biophysics Research Division, Department of Physics, Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Bandung 40132, Indonesia

20230527_v5| https://doi.org/10.5281/zenodo.7976249

Silakan berdiskusi untuk kuliah hari ini di
https://github.com/dudung/sk5003-02-2022-2/issues/12

Kerangka

SAP dan referensi

Minggu 8

Minggu	Topik	Subtopik	Capaian Belajar
11	Model komputasi fundamental dengan Python	Model empirik dengan interpolasi dan fitting kurva, penggunaan array dengan Numpy	Kemampuan untuk membuat model empirik dengan interpolasi dan fitting kurva, menggunakan array dengan Numpy

Referensi utama

- Jose M. Garrido, "Introduction to Computational Models with Python", Routledge, 1st edition, 2020,

  url https://isbnsearch.org/isbn/9780367575533.

R1

C16

- Interpolation
- Curve fitting

C17

- Vector
- Addition
- Multiplication
- Dot product
- Cross product

Model empirik

Model empirik

- Empirical models are only supported by experimental data, while the fundamentals and mechanisms underlying processes in a system are not considered.
- Developing an empirical model is a common methodology used to derive a direct correlation between inputs and outputs of a system, especially when it is difficult or impossible to develop a comparable mathematical model.

-, "Empirical model", -, url https://www.sciencedirect.com/topics/engineering/empirical-model [20230527].

Model empirik (lanj.)

- Empirical model can be used to model a system based on their underlying assumptions.
- Empirical models offer simplistic solutions for quantitative comparisons between different operating conditions.
- An empirical model can provide reliable results when it is based on a substantial amount of test data. However, the process of conducting a large number of tests, in particular system, is often costly and impractical.

Model empirik (lanj.)

- Empirical models are based on correlations obtained from analysis of experimental data.
- Empirical models are not based on any specific theory of an operation and are derived by fitting models to experimental data.
- Empirical models that have been used for the handling of some data have typically used curve fitting processes to generalise the results of experiments.

Model empirik (lanj.)

- Empirical models use relationships between X and Y measurements to estimate Z, where $Z = f(X, Y)$.

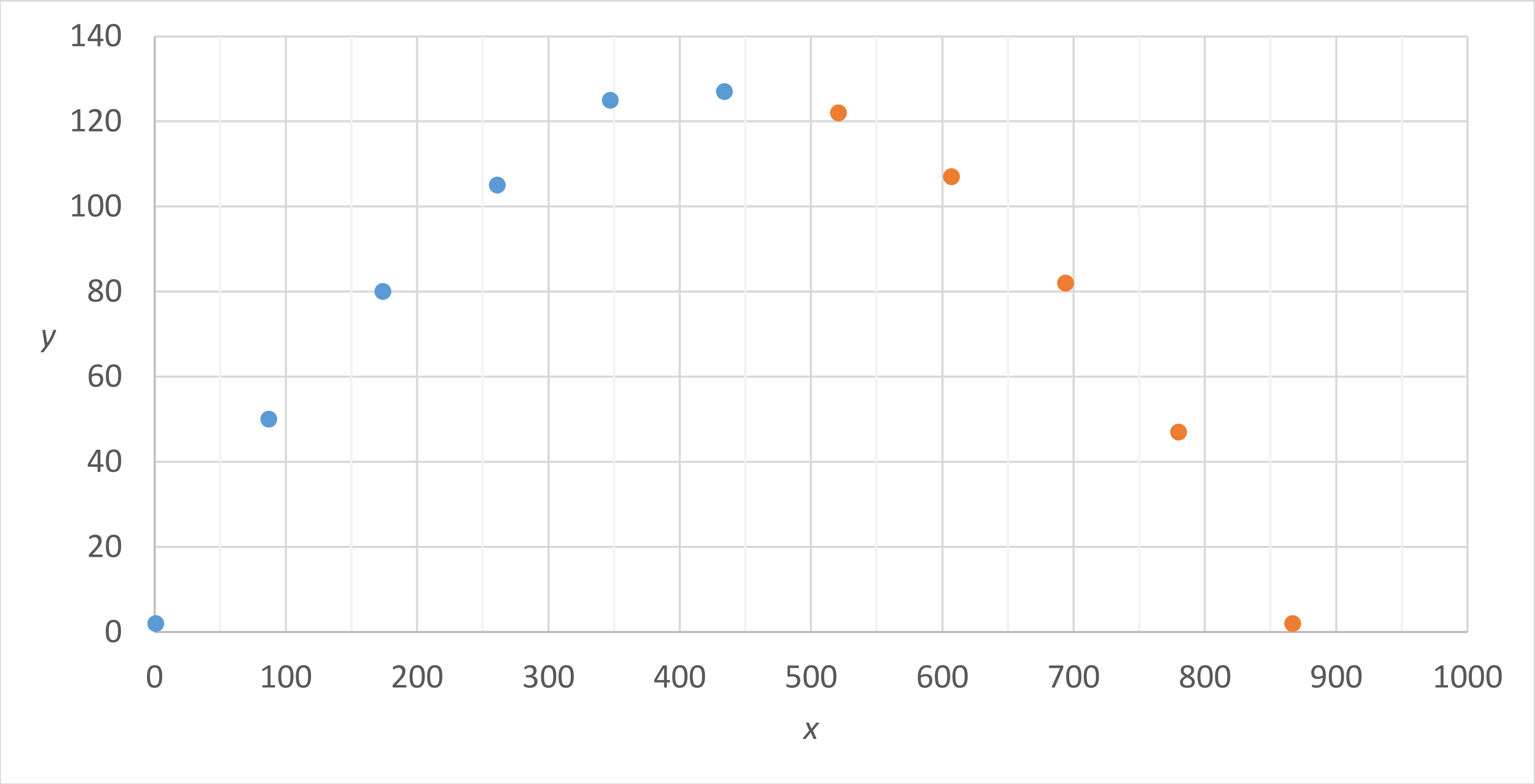
Model matematik

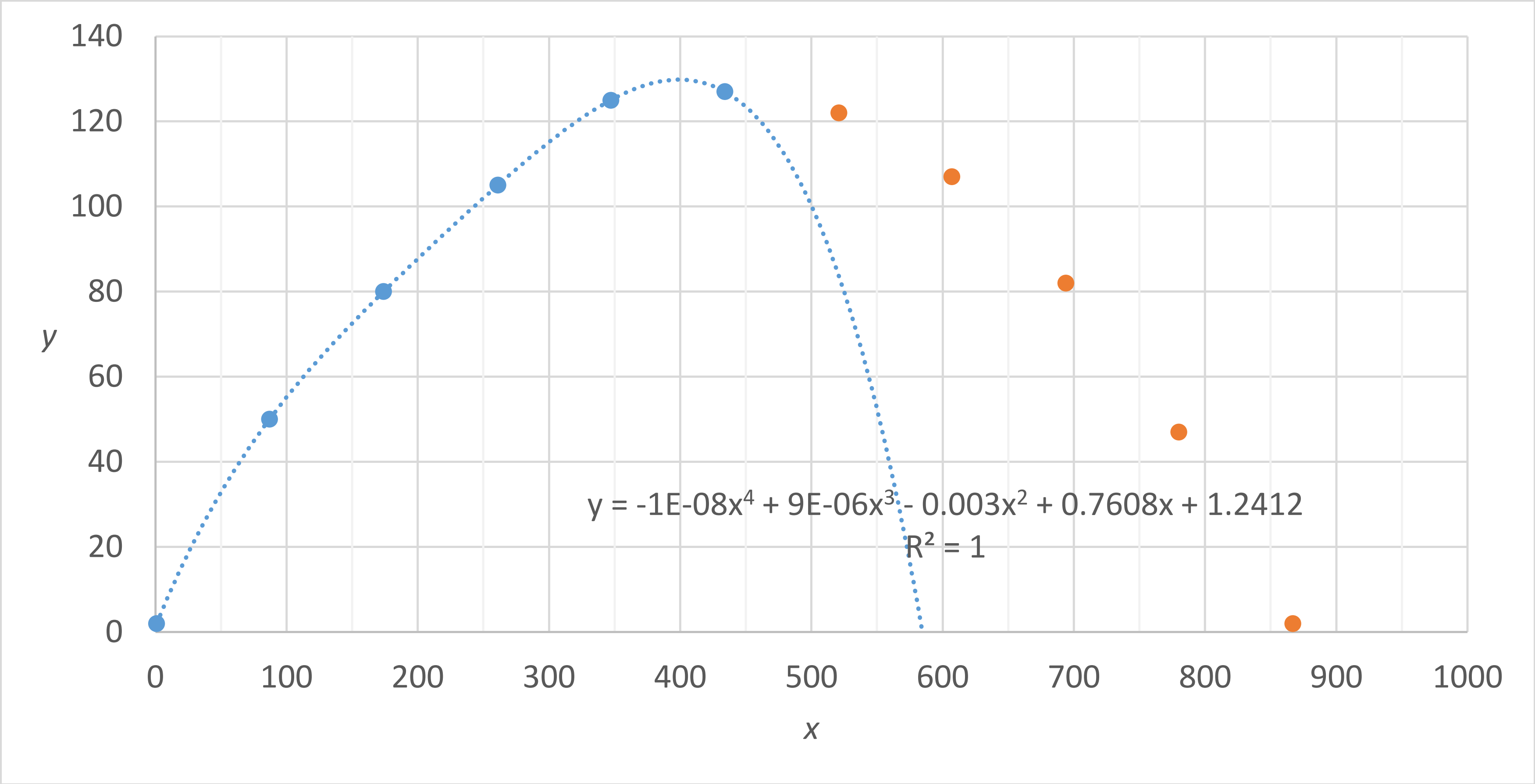
Model matematik

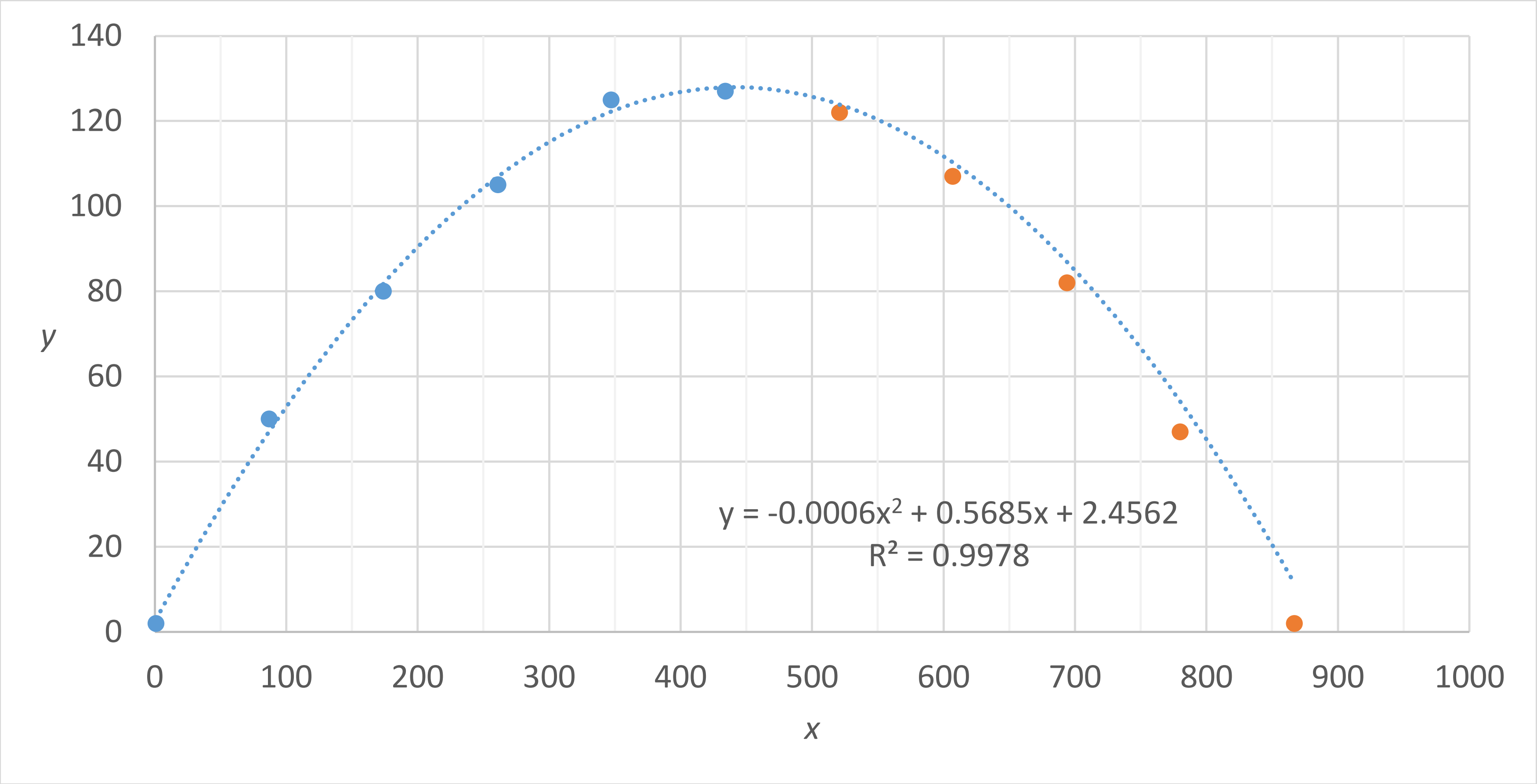
- Memiliki konsep atau teori sebagai dasarnya.
- Parameter-parameter yang digunakan saat melakukan fitting data dengan kurva akan “berbicara”.
- Hasil belum tentu akurat secara model, tetapi secara teori lebih tepat.

Gerak parabola

- Materi fisika dasar yang banyak dimanfaatkan, terutama dalam perang (sains tidak berpihak, tergantung yang memanfaatkanya ☹).
- Dapat ditingkatkan kompleksitasnya dengan menambahkan gesekan udara dan adanya angin yang berubah-ubah arah dan besarnya.
- Ilustrasi yang digunakan masih tanpa angin dan tanpa gesekan udara.


140
120
100
80
60
40
20
0
y
0
100
200
300
400
500
600
700
800
900
1000
x


140
120
100
80
60
40
20
0
y
0
100
200
300
400
500
600
700
800
900
1000
x
y = -1E-08x4 + 9E-06x3 - 0.003x2 + 0.7608x + 1.2412
R² = 1


140
120
100
80
60
40
20
0
y
0
100
200
300
400
500
600
700
800
900
1000
x
y = -0.0006x² + 0.5685x + 2.4562
R² = 0.9978

Formula

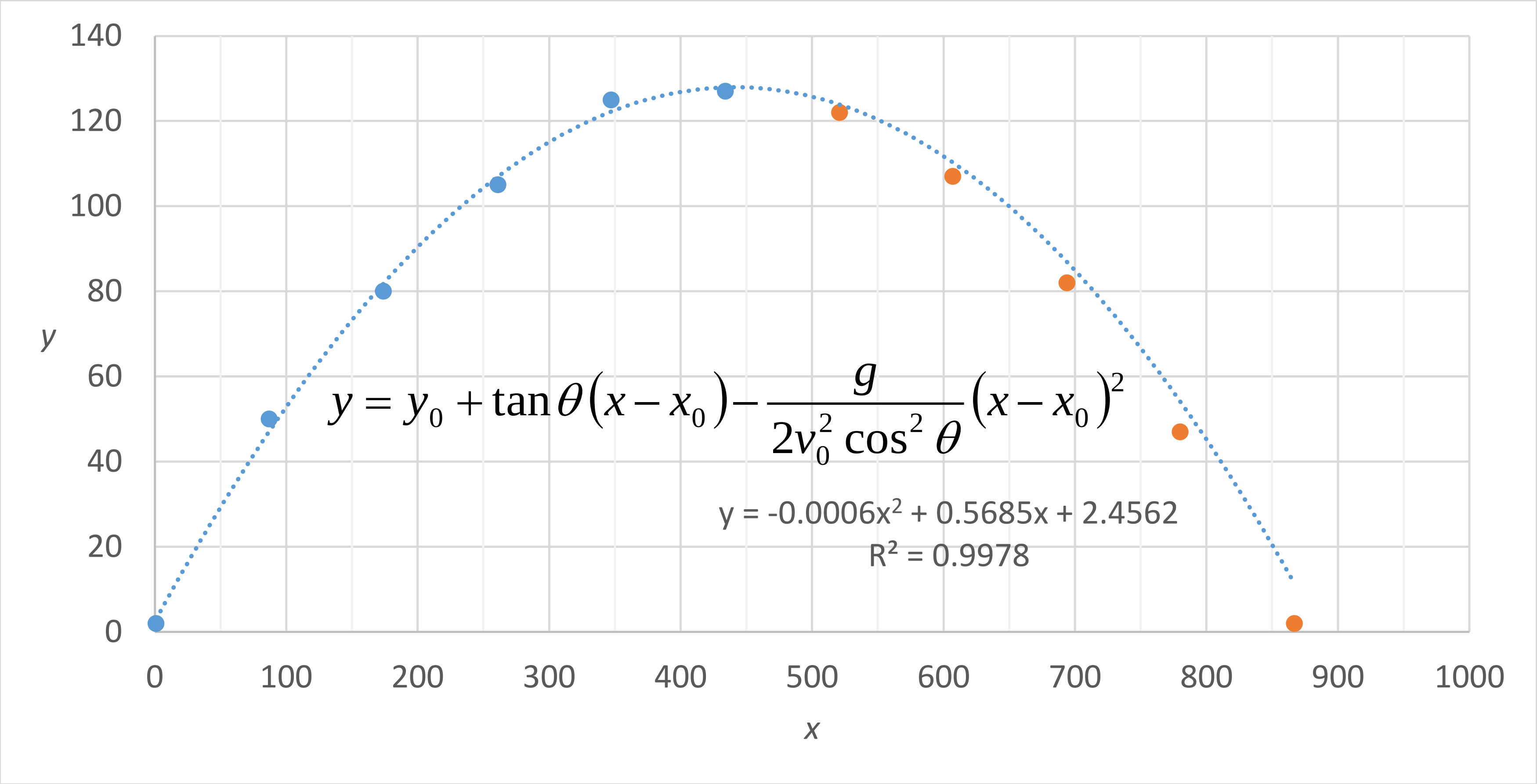
- Fungsi parametrik (dari kinematika)

$$x = x_0 + \left(v_0 \cos\theta\right)t$$

$$y = y_0 + \left(v_0 \sin\theta\right)t - \tfrac{1}{2} g t^2$$

- Posisi vertikal sebagai fungsi posisi horizontal

$$y = y_0 + \tan\theta\left(x - x_0\right) - \frac{g}{2v_0^2 \cos^2\theta}\left(x - x_0\right)^2$$



$$y = y_0 + \tan\theta\left(x - x_0\right) - \frac{g}{2v_0^2\cos^2\theta}\left(x - x_0\right)^2$$

Dari formula dan grafik

$$\frac{g}{2v_0^2\cos^2\theta} = 0.0006$$

$$\tan\theta + \frac{gx_0}{v_0^2\cos^2\theta} = 0.5685$$

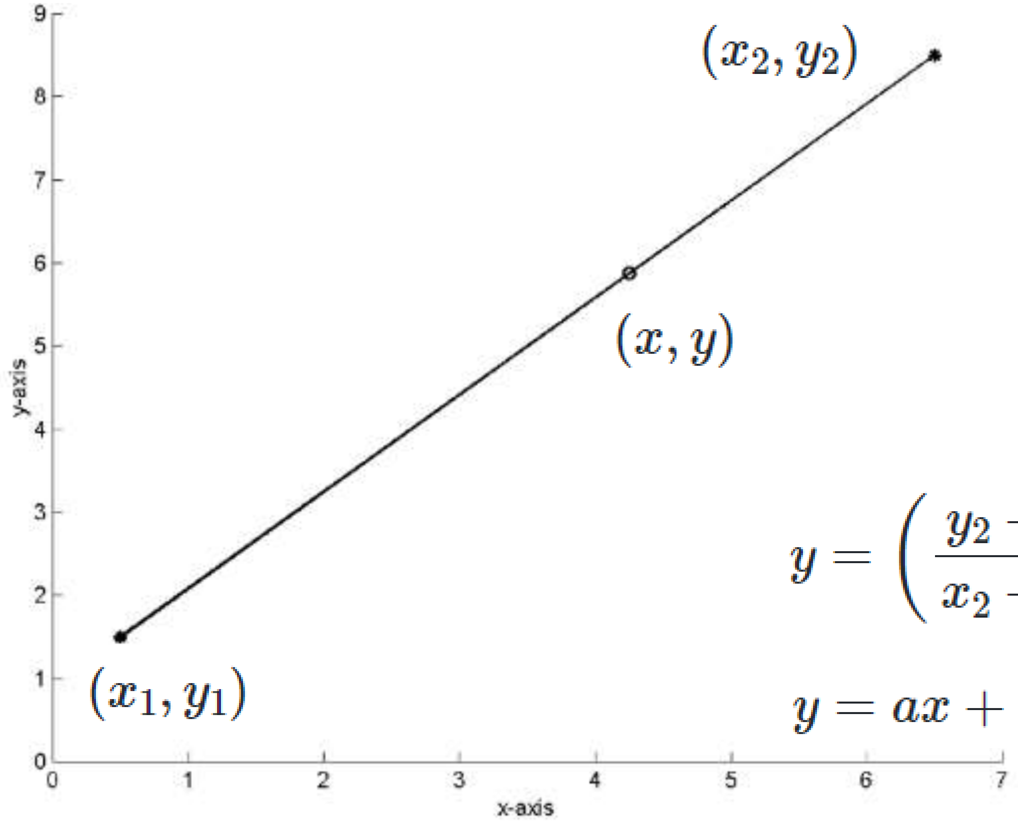
$$y = y_0 - \tan\theta\, x_0 - \frac{g}{2v_0^2\cos^2\theta}x_0^2 = 2.4562$$

Interpolasi

Interpolasi

- Estimasi data antara
- Interpolasi linier
- Interpolasi spline kubik
- Interpolasi polinomial Lagrange

Estimasi data antara



$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}$$

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$

$$y = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)x + \left[y_1 - x_1\left(\frac{y_2 - y_1}{x_2 - x_1}\right)\right]$$

$$y = ax + b$$

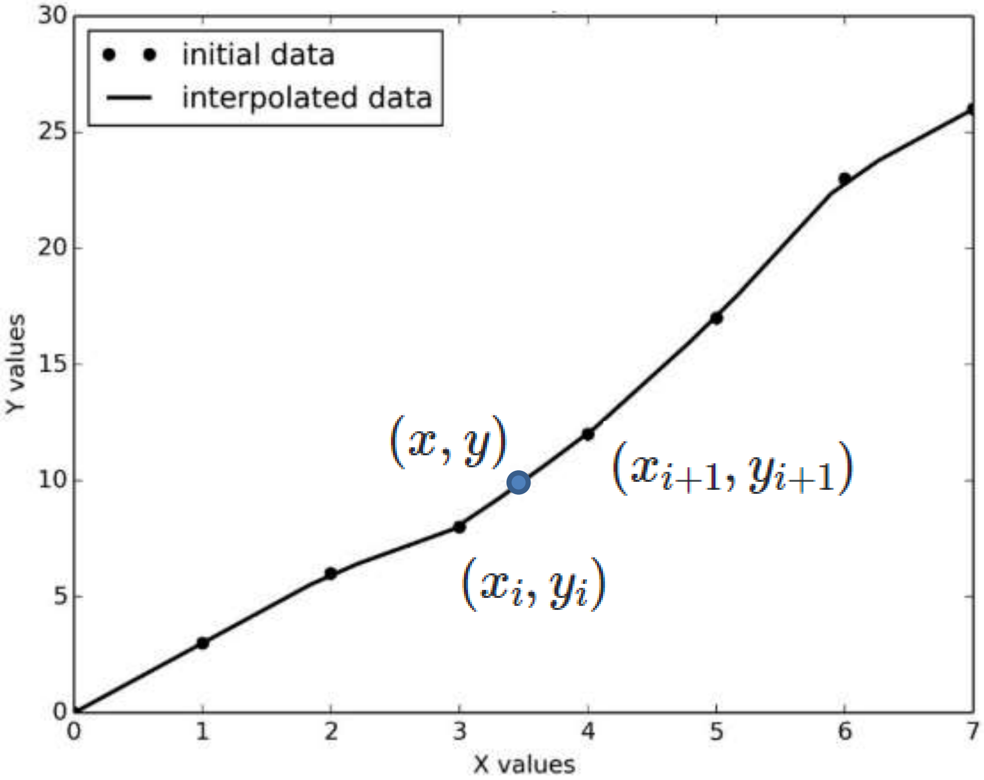
Interpolasi linier

- Membuat fungsi linier di antara dua titik data.

$$y = \left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right) (x - x_i) + y_i$$

$$i = 1 + \left\lfloor \frac{x - x_1}{\Delta x} \right\rfloor$$

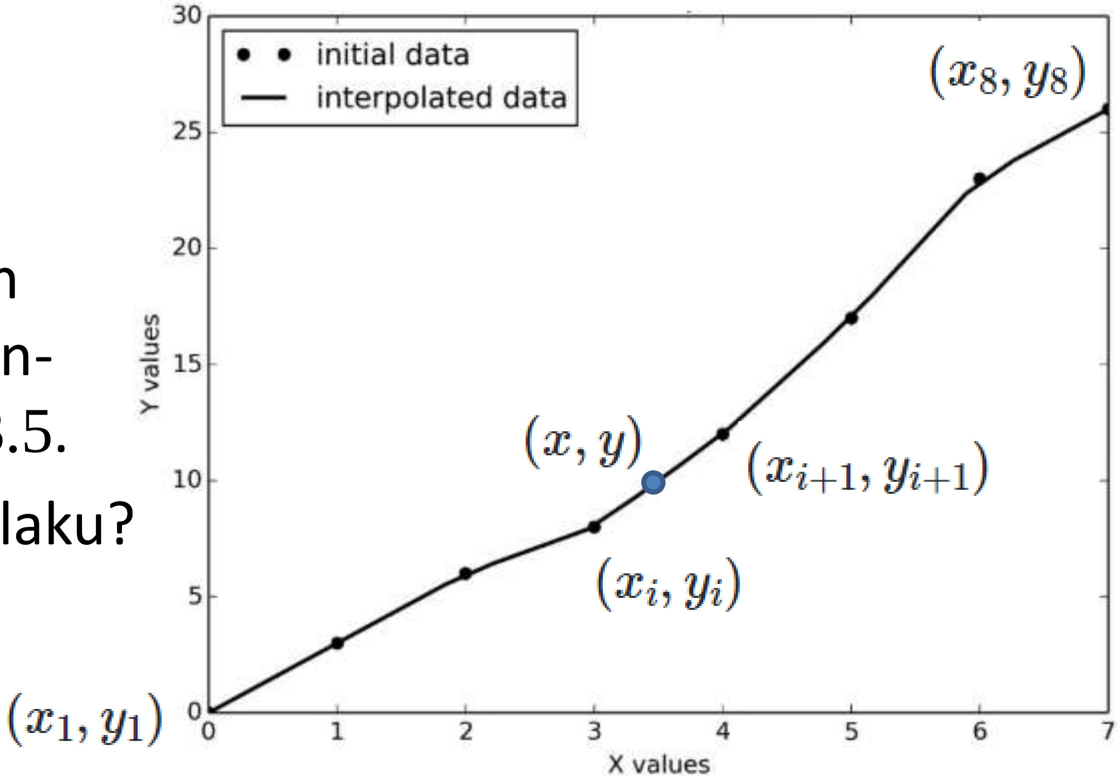
- Coba tunjukkan keberlakuannya.



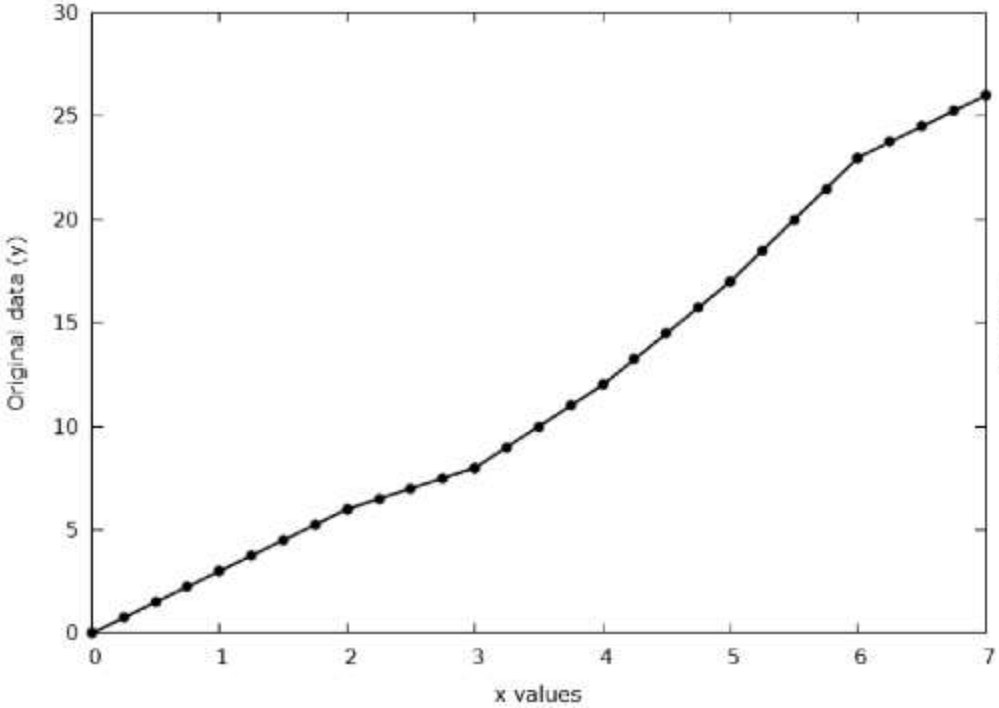
Interpolasi linier (lanj.)

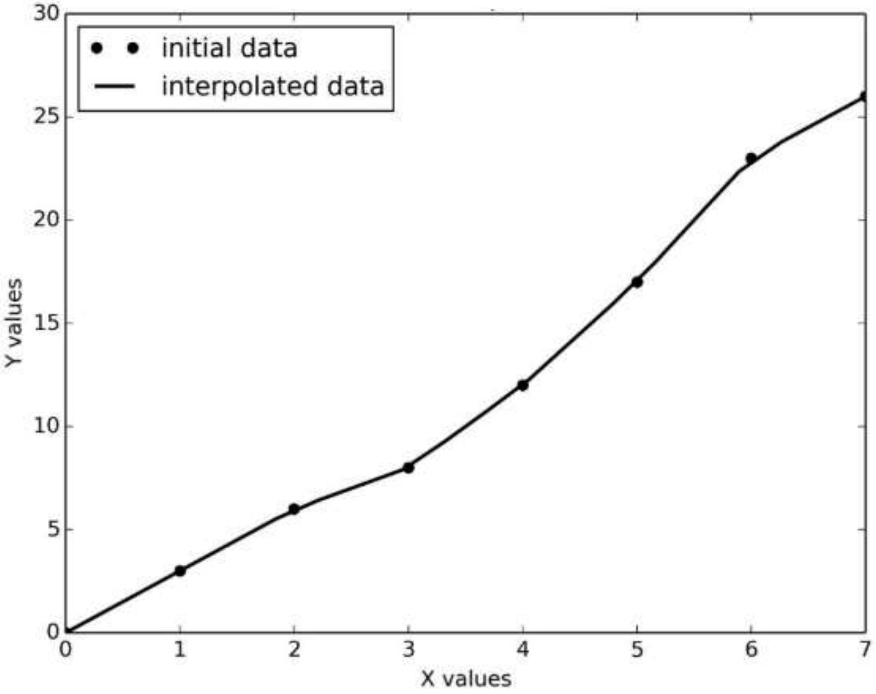
$$i = 1 + \left\lfloor \frac{x - x_1}{\Delta x} \right\rfloor$$

- Dengan menggunakan rumus sebelumnya tentukanlah i untuk $x = 3.5$.
- Apakah rumus itu berlaku?

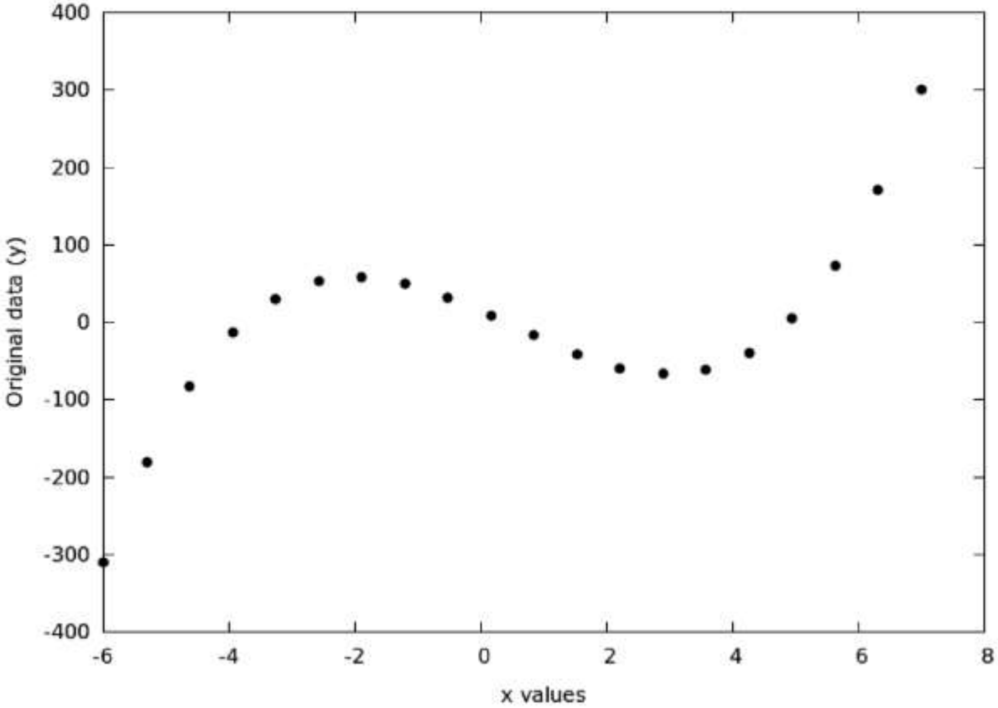


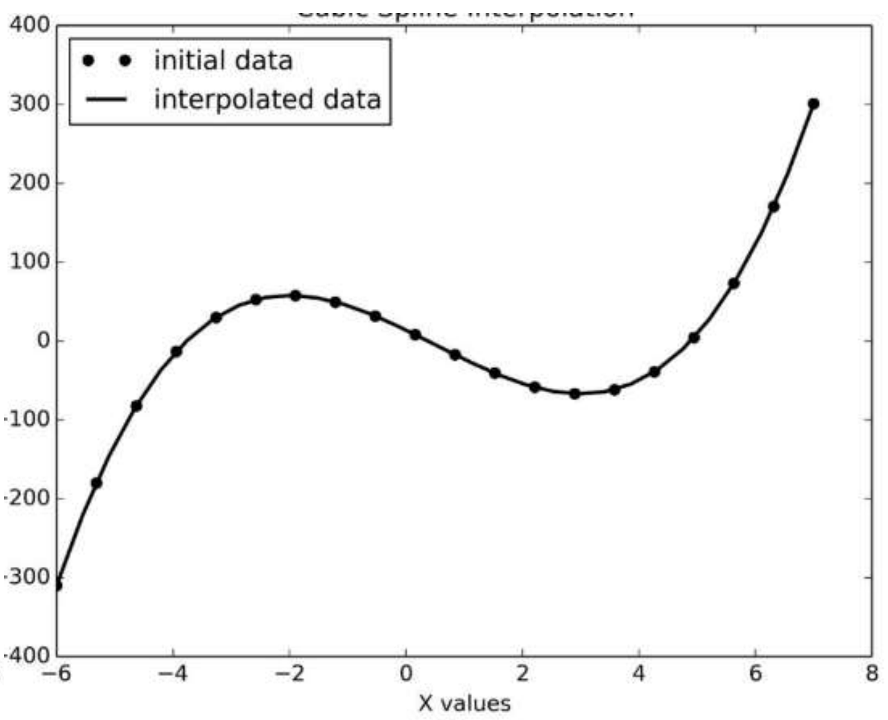
Interpolasi linier (lanj.)





Interpolasi spline kubik





Interpolasi spline kubik (lanj.)

- Untuk setiap rentang memenuhi

$$f(x) = \begin{cases} a_1 x^3 + b_1 x^2 + c_1 x + d_1 & \text{if } x \in [x_1, x_2] \\ a_2 x^3 + b_2 x^2 + c_2 x + d_2 & \text{if } x \in (x_2, x_3] \\ \dots \\ a_n x^3 + b_n x^2 + c_n x + d_n & \text{if } x \in (x_n, x_{n+1}] \end{cases}$$

Timo Denk, “Cubic Spline Interpolation”, 17 Jun 2017, url https://timodenk.com/blog/cubic-spline-interpolation/ [20230526].

Interpolasi spline kubik (lanj.)

$$
\begin{aligned}
f_1(x_1) &= y_1 \\
f_1(x_2) &= y_2 \\
f_2(x_2) &= y_2 \\
f_2(x_3) &= y_3 \\
&\cdots \\
f_n(x_n) &= y_n \\
f_n(x_{n+1}) &= y_{n+1} \, ,
\end{aligned}
$$

$$
\begin{aligned}
a_1 x_1^3 + b_1 x_1^2 + c_1 x_1 + d_1 &= y_1 \\
a_1 x_2^3 + b_1 x_2^2 + c_1 x_2 + d_1 &= y_2 \\
a_2 x_2^3 + b_2 x_2^2 + c_2 x_2 + d_2 &= y_2 \\
a_2 x_3^3 + b_2 x_3^2 + c_2 x_3 + d_2 &= y_3 \\
&\cdots \\
a_n x_n^3 + b_n x_n^2 + c_n x_n + d_n &= y_n \\
a_n x_{n+1}^3 + b_n x_{n+1}^2 + c_n x_{n+1} + d_n &= y_{n+1} \, ,
\end{aligned}
$$

Interpolasi spline kubik (lanj.)

$$\frac{d}{dx} f_1(x) = \frac{d}{dx} f_2(x) \qquad \big|_{x=x_2}$$
$$\frac{d}{dx} f_2(x) = \frac{d}{dx} f_3(x) \qquad \big|_{x=x_3}$$
$$\cdots$$
$$\frac{d}{dx} f_{n-1}(x) = \frac{d}{dx} f_n(x) \qquad \big|_{x=x_n} .$$

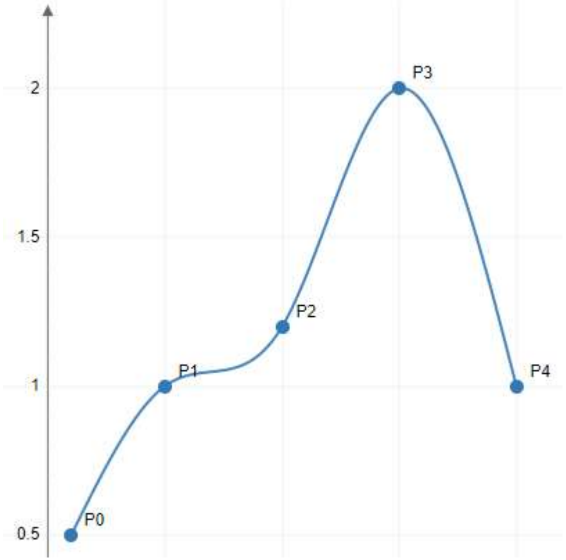
$$3a_1 x_2^2 + 2b_1 x_2 + c_1 = 3a_2 x_2^2 + 2b_2 x_2 + c_2$$
$$3a_2 x_3^2 + 2b_2 x_3 + c_2 = 3a_3 x_3^2 + 2b_3 x_3 + c_3$$
$$\cdots$$
$$3a_{n-1} x_n^2 + 2b_{n-1} x_n + c_{n-1} = 3a_n x_n^2 + 2b_n x_n + c_n .$$

Interpolasi spline kubik (lanj.)

x	y
1	0.5
5	1
10	1.2
15	2
20	1

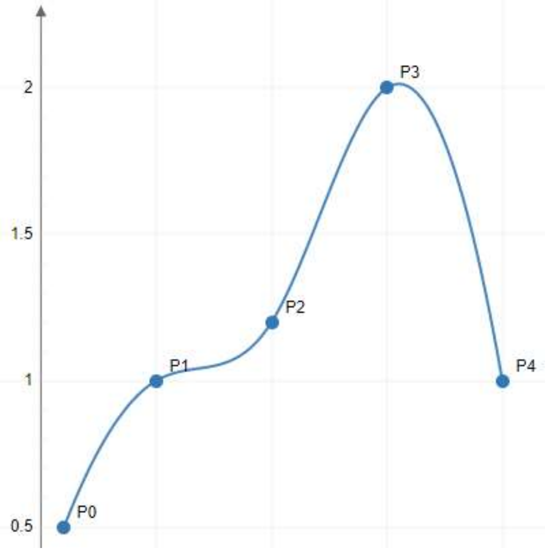
Interpolasi spline kubik (lanj.)

- Natural $$f(x) = \begin{cases} -2.1150 \cdot 10^{-3} \cdot x^3 + 6.3450 \cdot 10^{-3} \cdot x^2 + 1.5249 \cdot 10^{-1} \cdot x + 3.4327 \cdot 10^{-1}, & \text{if } x \in [1, 5], \\ 4.3832 \cdot 10^{-3} \cdot x^3 - 9.1128 \cdot 10^{-2} \cdot x^2 + 6.3986 \cdot 10^{-1} \cdot x - 4.6900 \cdot 10^{-1}, & \text{if } x \in (5, 10], \\ -6.9640 \cdot 10^{-3} \cdot x^3 + 2.4929 \cdot 10^{-1} \cdot x^2 - 2.7643 \cdot x + 1.0878 \cdot 10^{1}, & \text{if } x \in (10, 15], \\ 4.2728 \cdot 10^{-3} \cdot x^3 - 2.5637 \cdot 10^{-1} \cdot x^2 + 4.8205 \cdot x - 2.7046 \cdot 10^{1}, & \text{if } x \in (15, 20]. \end{cases}$$



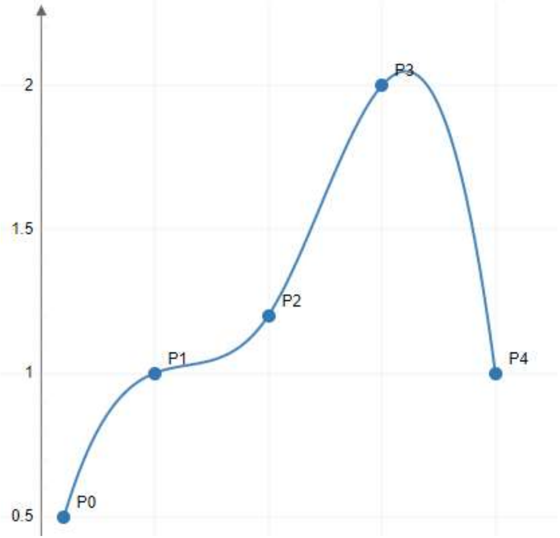
Interpolasi spline kubik (lanj.)

- Quadratic $f(x) = \begin{cases} -1.0089 \cdot 10^{-64} \cdot x^3 - 1.9656 \cdot 10^{-2} \cdot x^2 + 2.4294 \cdot 10^{-1} \cdot x + 2.7672 \cdot 10^{-1}, & \text{if } x \in [1,5], \\ 3.6763 \cdot 10^{-3} \cdot x^3 - 7.4802 \cdot 10^{-2} \cdot x^2 + 5.1866 \cdot 10^{-1} \cdot x - 1.8282 \cdot 10^{-1}, & \text{if } x \in (5,10], \\ -5.7191 \cdot 10^{-3} \cdot x^3 + 2.0706 \cdot 10^{-1} \cdot x^2 - 2.3000 \cdot x + 9.2126, & \text{if } x \in (10,15], \\ 0.0000 \cdot x^3 - 5.0298 \cdot 10^{-2} \cdot x^2 + 1.5604 \cdot x - 1.0089 \cdot 10^{1}, & \text{if } x \in (15,20]. \end{cases}$



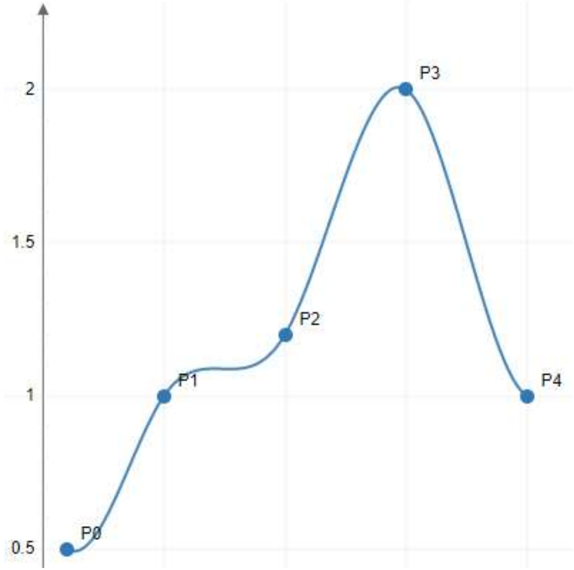
Interpolasi spline kubik (lanj.)

- Not-a-knot $f(x) = \begin{cases} 2.8222 \cdot 10^{-3} \cdot x^3 - 5.4600 \cdot 10^{-2} \cdot x^2 + 3.6511 \cdot 10^{-1} \cdot x + 1.8667 \cdot 10^{-1}, & \text{if } x \in [1,5], \\ 2.8222 \cdot 10^{-3} \cdot x^3 - 5.4600 \cdot 10^{-2} \cdot x^2 + 3.6511 \cdot 10^{-1} \cdot x + 1.8667 \cdot 10^{-1}, & \text{if } x \in (5,10], \\ -4.4044 \cdot 10^{-3} \cdot x^3 + 1.6220 \cdot 10^{-1} \cdot x^2 - 1.8029 \cdot x + 7.4133, & \text{if } x \in (10,15], \\ -4.4044 \cdot 10^{-3} \cdot x^3 + 1.6220 \cdot 10^{-1} \cdot x^2 - 1.8029 \cdot x + 7.4133, & \text{if } x \in (15,20]. \end{cases}$

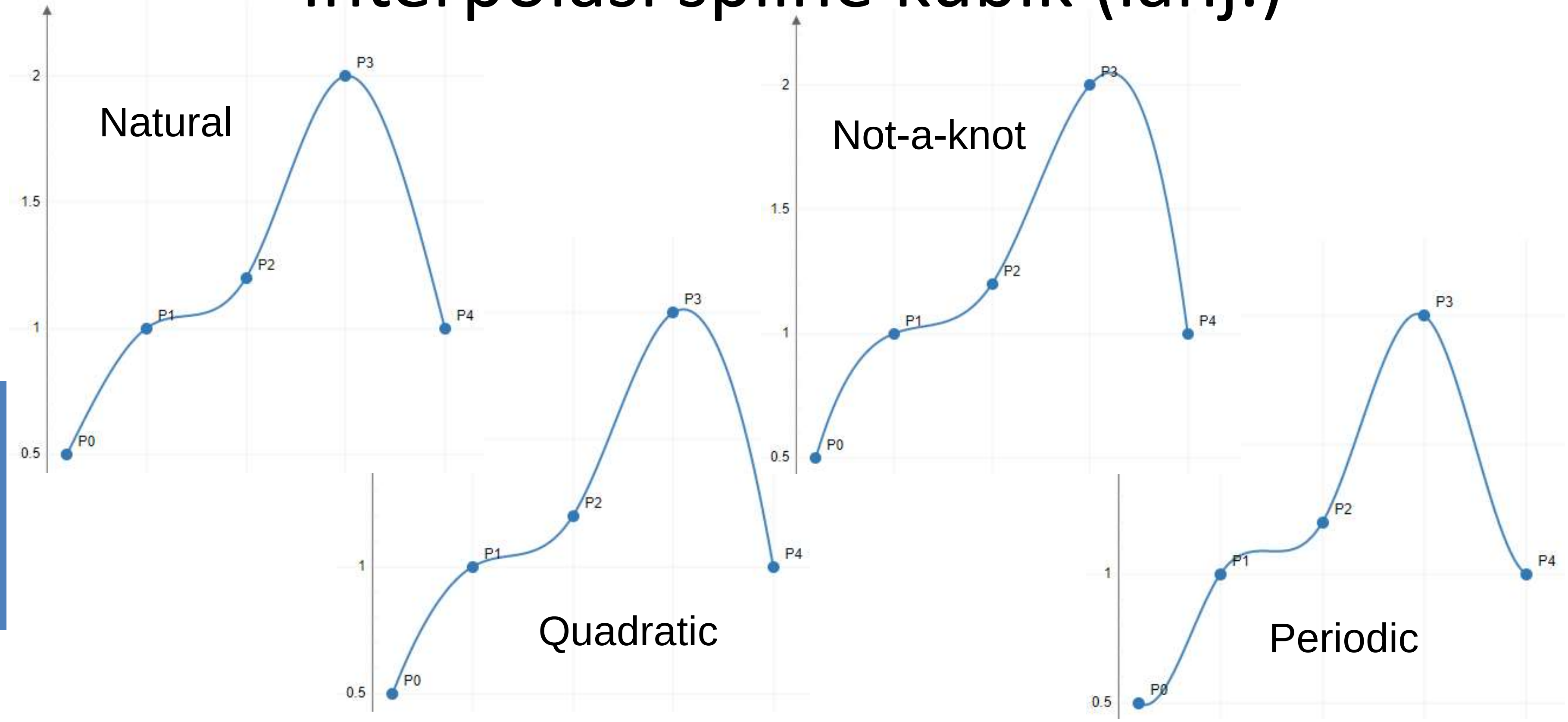


Interpolasi spline kubik (lanj.)

- Periodic $f(x) = \begin{cases} -1.1554 \cdot 10^{-2} \cdot x^3 + 1.2456 \cdot 10^{-1} \cdot x^2 - 2.6417 \cdot 10^{-1} \cdot x + 6.5117 \cdot 10^{-1}, & \text{if } x \in [1,5], \\ 6.7564 \cdot 10^{-3} \cdot x^3 - 1.5010 \cdot 10^{-1} \cdot x^2 + 1.1091 \cdot x - 1.6376, & \text{if } x \in (5,10], \\ -9.4811 \cdot 10^{-3} \cdot x^3 + 3.3703 \cdot 10^{-1} \cdot x^2 - 3.7621 \cdot x + 1.4600 \cdot 10^{1}, & \text{if } x \in (10,15], \\ 1.1968 \cdot 10^{-2} \cdot x^3 - 6.2818 \cdot 10^{-1} \cdot x^2 + 1.0716 \cdot 10^{1} \cdot x - 5.7790 \cdot 10^{1}, & \text{if } x \in (15,20]. \end{cases}$



Interpolasi spline kubik (lanj.)



Polinomial interpolasi Lagrange

$$L(x) = \sum_{j=0}^{k} y_j \ell_j(x).$$

$$\ell_j(x) = \frac{(x - x_0)}{(x_j - x_0)} \cdots \frac{(x - x_{j-1})}{(x_j - x_{j-1})} \frac{(x - x_{j+1})}{(x_j - x_{j+1})} \cdots \frac{(x - x_k)}{(x_j - x_k)}$$

$$= \prod_{\substack{0 \le m \le k \\ m \ne j}} \frac{x - x_m}{x_j - x_m}.$$

Wikipedia contributors, 'Lagrange polynomial', Wikipedia, The Free Encyclopedia, 12 April 2023, 16:12 UTC, url https://en.wikipedia.org/w/index.php?oldid=1149496043 [20230526].

NumPy

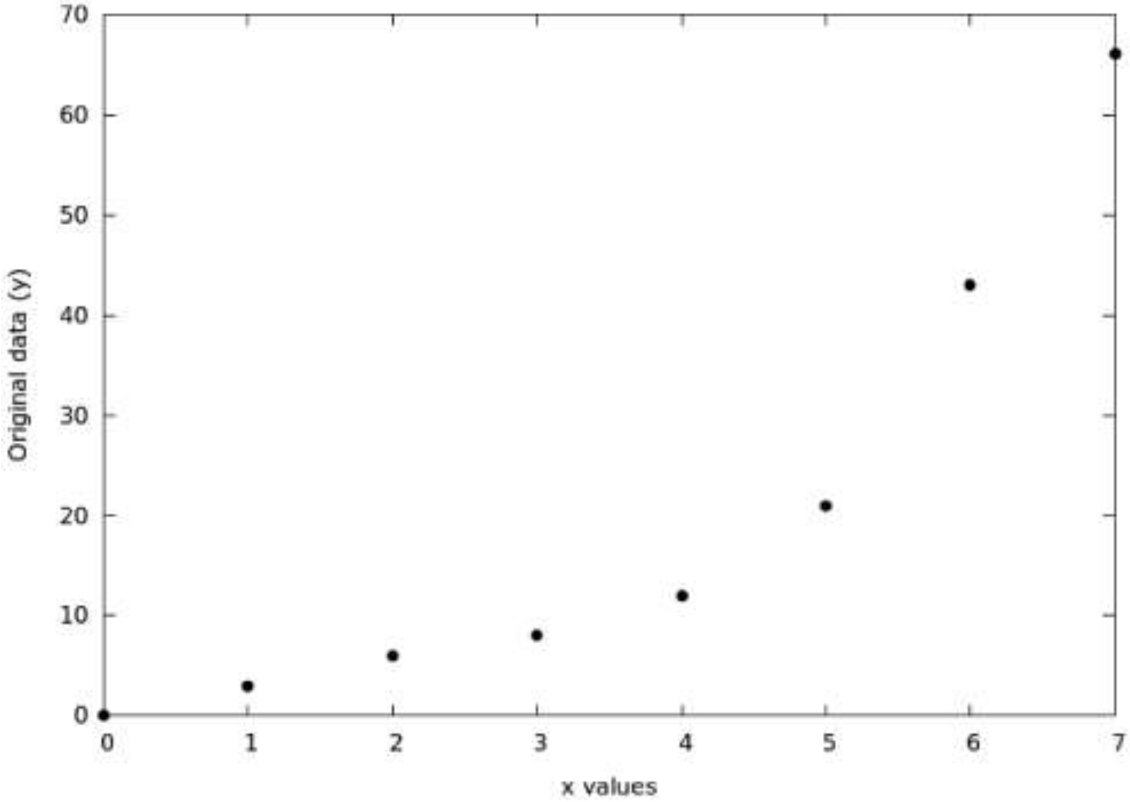
```
# Program that performs linear interpolation
import numpy as np

y = [0.0, 3.0, 6.0, 8.0, 12.0, 17.0, 23.0, 26]
y = np.array(y)
print "Values of array y:"
print y
xi = 0.0
xf = 7.0
M = np.size(y)
x = x = np.linspace(xi, xf, M)
print "Values of array x:"
print x
# generate an array of 20 intermediate points
N = 20
xint = np.linspace(xi, xf, N)
yint = np.interp(xint, x, y)
print "Interpotated values of x and y:"
for j in range(N):
    print xint[j], yint[j]
```

NumPy + SciPy

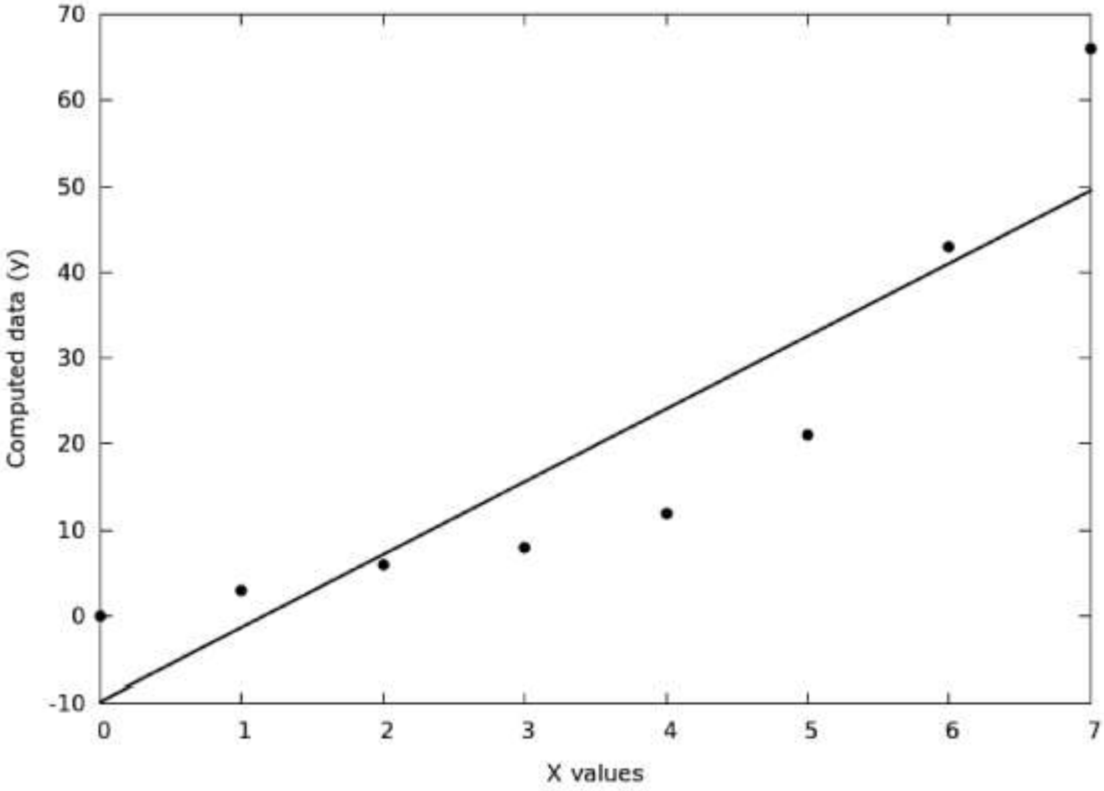
```
# Cubic spline interpolation
#File: csinterp.py  Sep 2, 2014
import numpy as np
from scipy import interpolate
y = [-310.0, -179.8, -82.3, -13.6, 30.0, 52.6,57.8, 49.6,
 31.8, 8.2, -17.2,-40.8, -58.6, -66.8, -61.5,-39.0, 4.6,
 73.3, 170.8, 301.0]
y = np.array(y)
N = np.size(y)
xi = -6.0
xf = 7.0
x = np.linspace(xi, xf, N)
sprep = interpolate.splrep(x, y, s=0) # spline of y
M = int(1.5 * N)  # more points
xint = np.linspace(xi, xf, M)
yint = interpolate.splev(xint, sprep, der=0) # interp
print "Cubic Spline interpolation:"
for j in range(M):
      print xint[j], yint[j]
```

Fitting kurva (curve fitting)

Data



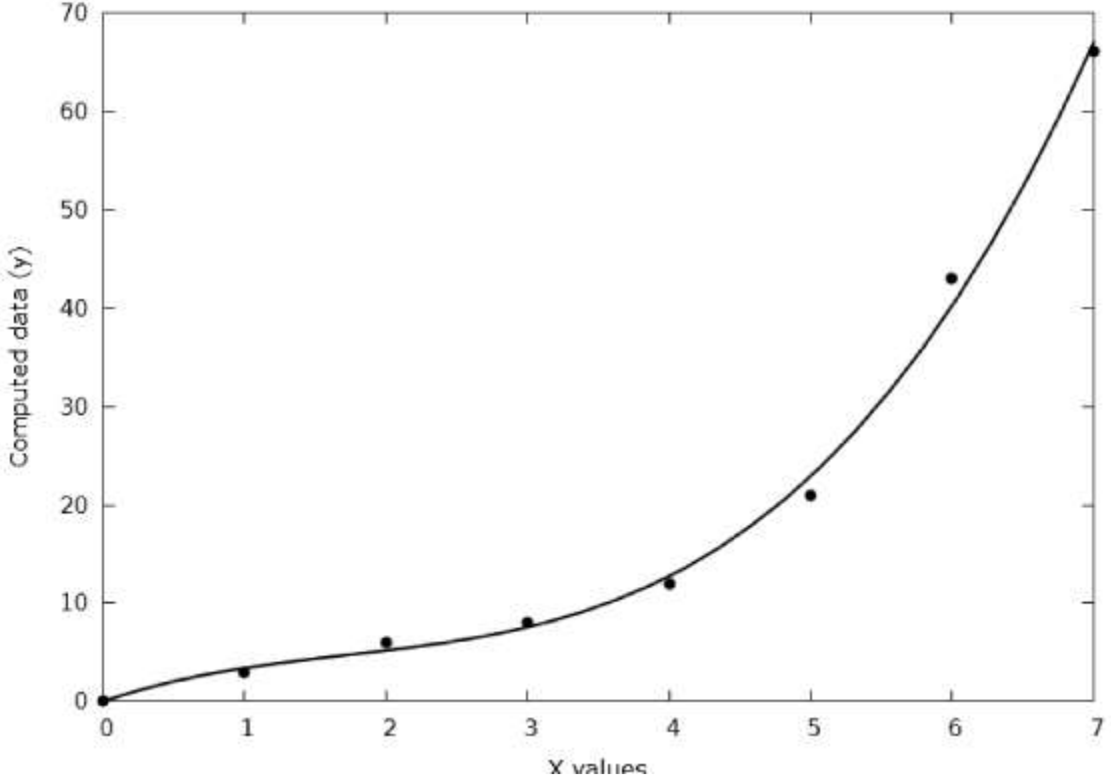
Fitting dengan fungsi linier (orde 1)



NumPy

```
M = 30      # number of data points to compute
xi = 0.0    # first value of x
xf = 7.0    # final value of x
y = [0.0, 3.0, 6.0, 8.0, 12.0, 21.0, 43.0, 66.0]
y = np.array(y)
sn = np.size(y) # number of points
x = np.linspace(xi, xf, sn)
deg = 1  # degree of polynomial function
print "Values of X and Y:"
for j in range (sn):
   print x[j], y[j]
c = np.polyfit(x, y, deg)
print "Coefficient list:"
print c
# Evaluate polynomial
xc = np.linspace(xi, xf, M) # new x points
yc = np.polyval(c, xc) # new y points
print "Evaluation of polynomial"
for j in range(M):
      print xc[j], yc[j]
```

Fitting dengan fungsi linier (orde 3)



NumPy

```python
M = 30       # number of data points to compute
xi = 0.0     # first value of x
xf = 7.0     # final value of x
y = [0.0, 3.0, 6.0, 8.0, 12.0, 21.0, 43.0, 66.0]
y = np.array(y)
sn = np.size(y) # number of points
x = np.linspace(xi, xf, sn)
deg = 1  # degree of polynomial function
print "Values of X and Y:"
for j in range (sn):
   print x[j], y[j]
c = np.polyfit(x, y, deg)
print "Coefficient list:"
print c
# Evaluate polynomial
xc = np.linspace(xi, xf, M) # new x points
yc = np.polyval(c, xc) # new y points
print "Evaluation of polynomial"
for j in range(M):
      print xc[j], yc[j]
```

- Bagaimana kira-kira kodenya?
- Modifikasi `deg = 3` untuk orde 3.

Materi mandiri

Matriks dan vektor

- Matriks dan vektor pada matematika dapat dihitung dalam Python.
- Secara simbolik dapat menggunakan SymPy.
- Secara numerik dapat menggunakan NumPy.
- Beberapa operasinya adalah: penjumlahan / pengurangan, perkalian / pembagian, rotasi, proyeksi, besar, determinan, transpos, invers.

Catatan

- Pelajari materi yang termasuk pada bagian mandiri ini.
- Buat beberapa contoh dan selesaikan secara teori.
- Gunakan contoh yang sama dan selesaikan secara numerik dengan menggunakan NumPy.

Penutup

Pertanyaan

- Apakah perbedaan antara interpolasi linier, quadratik, kubik, kubik spline, polinomial Lagrange?
- Jelaskan perbedaan mendasar antara interpolasi dan fitting kurva?
- Apakah perbedaan antara model empirik dan matematik? Kapan digunakan yang pertama dan kapan yang kedua?
- Jelaskan simbol dan pemanfaatan fungsi floor dan ceil. Apakah digunakan dalam pertemuan ini? Bila ya, di bagian mana?

Terima kasih

-