ENGG1003 - Monday Week 6

Interpolation, Assignment 1 and Mid-term quiz

Steve Weller

University of Newcastle

29 March 2021

Last compiled: March 28, 2021 7:36pm +11:00

Lecture overview

1. Interpolation
2. Assignment 1
3. Mid-term quiz

The story so far

- variables and data types
- arrays (using `numpy`)
- plotting (using `matplotlib`)
- flow control
  - `if`
  - `while`
  - `for`
- functions

Most of ENGG1003 from here uses these elements of Python to solve Engineering problems

1) Interpolation

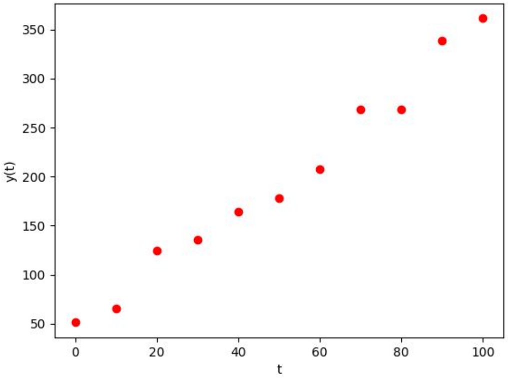
Two common forms of *curve-fitting* in Engineering applications:

1. *interpolation*
   - today's lecture
2. *regression*
   - considered in detail later in ENGG1003

- we now demonstrate both curve-fitting methods applied to the same dataset

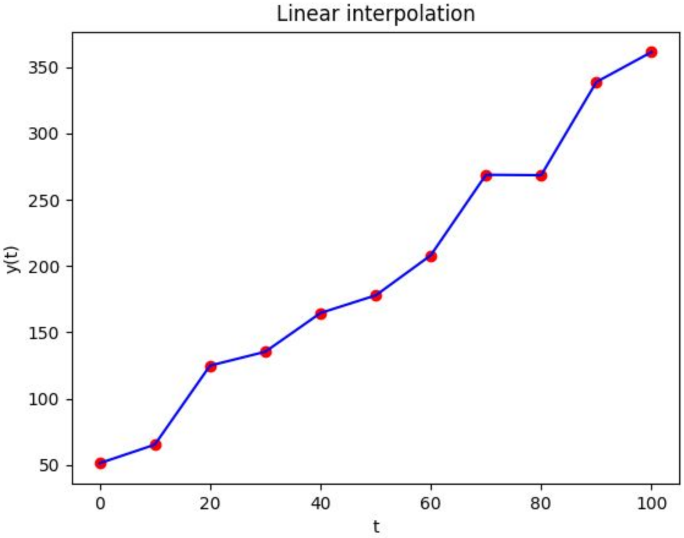
Curve-fitting dataset

`Week6Mon.py`

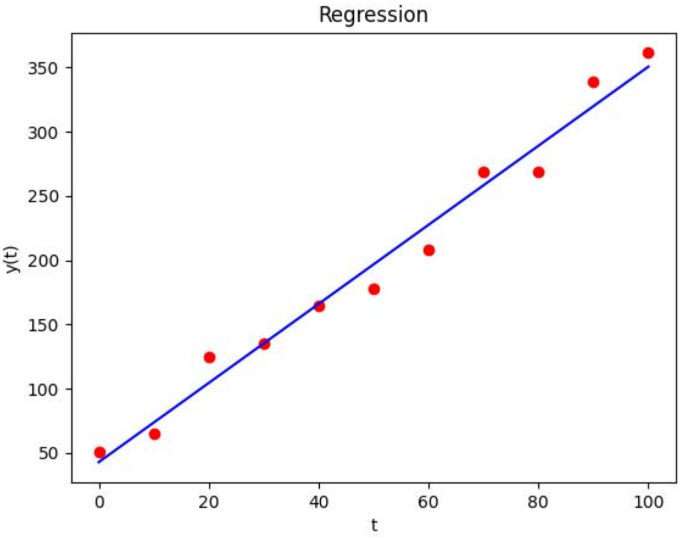


- 11 pairs of data points $(t_i, y_i), i = 0, 1, 2, \ldots, 10$
  $(0, 51.29), (10, 65.24), (20, 124.89), \ldots, (100, 361.32)$

Interpolation



Regression



Interpolation vs. regression

- **interpolation:** joining the dots
  - obtain value of y at some intermediate point

- **regression:** fitting a straight line
  - when there's "too much data", simplify
  - here, simplifying to a straight line
  - we return to choosing "best" straight line later in ENGG1003
  - no more regression in this lecture

- both interpolation & regression involve creating a function (blue line) from data (red dots)

Functions

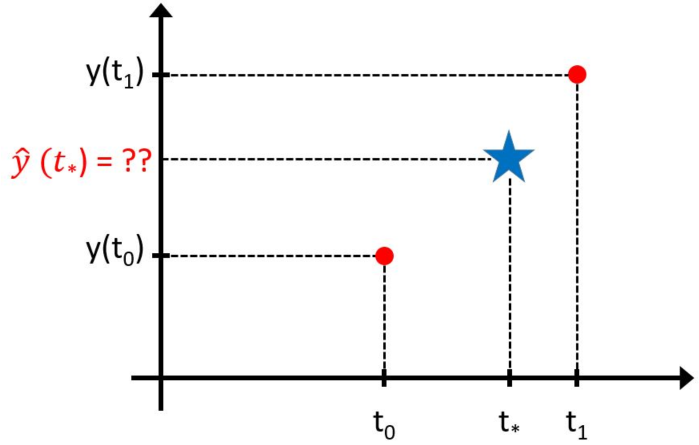
- in maths, you've seen functions written as follows:

$$y = f(t)$$

- function f, takes *argument* t and returns a *value* which is assigned to y

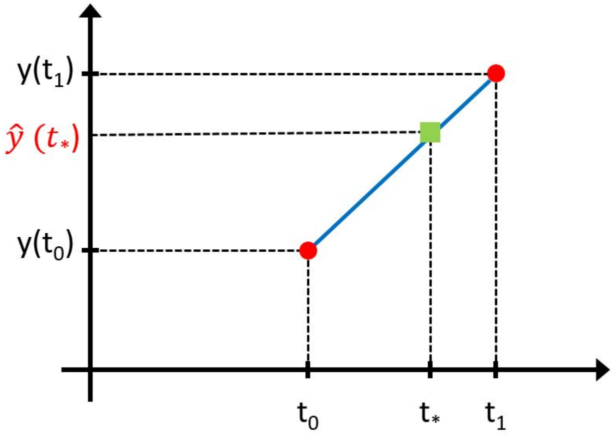
- . . . and last week we saw Python uses the same terminology with functions

The interpolation problem



Given: data points $(t_0, y(t_0))$ & $(t_1, y(t_1))$ and $t_\star$
Calculate: interpolated value $\hat{y}(t_\star)$

Linear interpolation



- interpolated value $\hat{y}(t_\star)$ lies on straight line connecting $(t_0, y(t_0))$ & $(t_1, y(t_1))$

Linear interpolation using `interp1d`

- `interp1d` function from `scipy.interpolate`
- `pip install scipy` at console first
- call to `interp1d` returns a function
- use the function in console
- live demo

Linear interpolation in Python

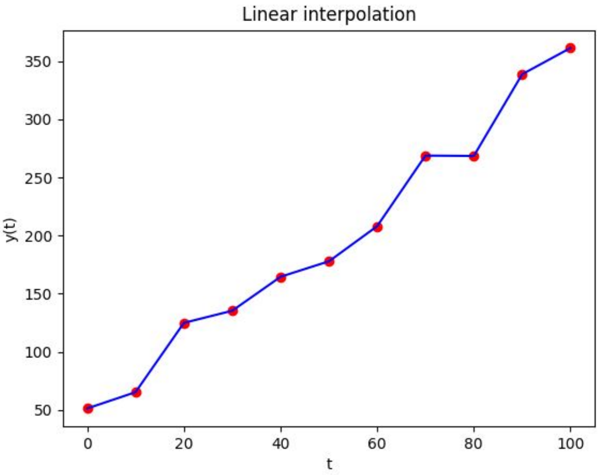
Week6MonLinear.py

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.interpolate import interp1d

# seed random number generator to reproduce lecture results
np.random.seed(27101967)

N = 11
t = np.linspace(0,100,11)           # 0,10,20,...,100
tnew = np.linspace(0,100,100*N)     # 0,0.1,0.2,...,100
n = np.random.uniform(-25,25,N)     # noise on linear function

m = 3                               # gradient
b = 50                              # intercept
y = m*t + b + n                   # dataset is straight line + noise
```

Linear interpolation in Python (ctd.)

```python
# INTERPOLATION
f = interp1d(t, y)

# PLOT RESULTS
plt.plot(t, y, 'ro')
plt.xlabel('t')
plt.ylabel('y(t)')
plt.plot(tnew, f(tnew), 'b')
plt.title('Linear interpolation')
plt.show()
```

Linear interpolation



- “stitches together” straight line segments

Beyond linear interpolation

Problem: slopes of adjacent straight lines change abruptly at data points

Cubic splines

- degree 1 is linear, degree 2 is parabola, degree 3 is cubic
- we're hiding lots of maths here, but we can use Python to implement without needing that maths
- maths is interesting!

Cubic spline in Python code

```python
# INTERPOLATION
f3 = interp1d(t, y, 'cubic')

# PLOT RESULTS
plt.plot(t, y, 'ro')
plt.xlabel('t')
plt.ylabel('y(t)')
plt.plot(tnew, f3(tnew), 'b')
plt.title('Cubic spline interpolation')
plt.show()
```

Linear interpolation:

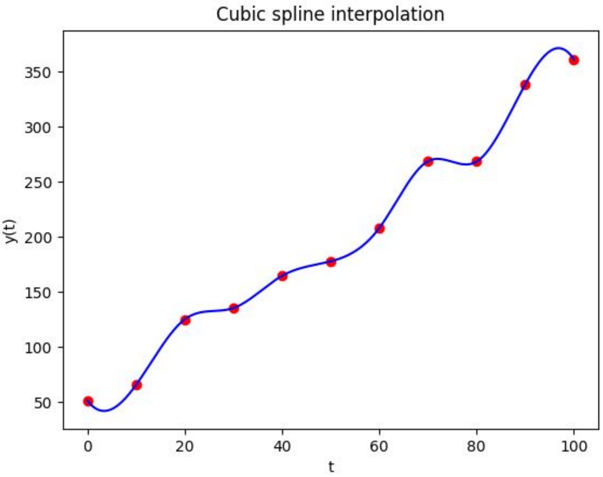
```python
f = interp1d(t, y)
```

Cubic spline interpolation:

```python
f3 = interp1d(t, y, 'cubic')
```

- Live demo of `Week6MonCubic.py`

Cubic spline interpolation



- "stitches together" *cubic* polynomials

2) Assignment 1

- released today, Monday 29 March 2021
- upload Python file to BB no later than 9:00 am, Monday 19 April
- counts for 20% of course grade
- each student will be assessed by demonstrator in week 7 face-face computer lab (after recess)

2) Assignment 1

- basic idea in assignment: analysis of GPS data
- in more detail:
  - GPS data collected during a mountain bike ride
  - GPS track log is presented in a `csv` file containing timestamps, latitude, longitude, and elevation columns recorded at 1 second intervals
  - project asks you to analyse GPS data, eg: fitness trackers
  - use interpolation where GPS signal was lost
- computer labs this week:
  - there is no week 6 lab sheet
  - get started on assignment in place of lab sheet

3) Mid-term quiz

- Thursday 1 April, 4–5pm
  - during scheduled lecture time
  - mid-term quiz instead of lecture this week so...
  - **NO Zoom or YouTube livestream on 1 April**
- 50-minute quiz
- open-book
- quiz will appear on BB at 4:05 pm

  ...and will disappear at 4:55 pm
- counts for 15% of course grade

What to expect, and how to prepare

- the quiz will ask you to write Python code to:
  - load a `csv` file
  - perform some calculations on a specified column of the file ie: array processing
    (each student gets assigned a unique column in the file)
- once your program is complete, you do two things:
  1. enter the results of your program into BB
  2. upload your Python code to BB
- **there is a practice quiz NOW on BB**
  - you can practice as many times as you like!
- live demo of how the mid-term quiz will run

Lecture summary

- Interpolation
  - linear interpolation
    - straight line “join the dots”
  - cubic spline interpolation
    - smoothly connects data points

- Assignment 1
  - released today
  - submissions due on BB by 9:00 am, Monday 19 April
  - worth 20%

- Mid-term quiz
  - **there is a practice quiz NOW on BB**
  - done on BB at 4:00 pm, Thursday 1 April
  - worth 15%