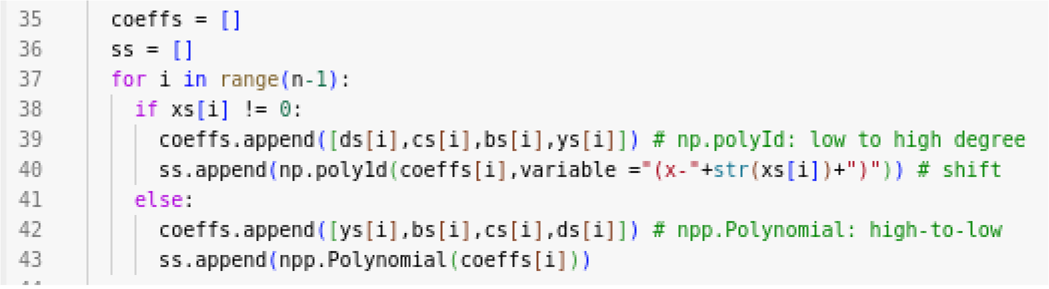
nmi : spring 2024 : quiz 05 : python, least squares, usw

q1. (5 pts) what is the maximum effect of all the extra credit in the world as applied to this class?

q2. (5 pts) what kind of matrices is QR-factorization useful for? (hint: remember the van der monde example. that was the one sensei went back and fixed bc she had the rows and colums swapped.)

q3. (5 pts) modified gram-schmidt improves on classical gram-schmidt and householder reflection improves on modified gram-schmidt, so why is modified gram-schmidt and not householder reflection the QR method in python libraries? (hint: its best to have the latest lecture 10 notebook, "nmi_10_0404_qr.ipynb". or prop up what you know with your bff google.)

q4. (5 pts) in the below code scrape excerpt from the cubic spline lecture, consider lines 39 and 42.



```
35    coeffs = []
36    ss = []
37    for i in range(n-1):
38      if xs[i] != 0:
39        coeffs.append([ds[i],cs[i],bs[i],ys[i]]) # np.polyId: low to high degree
40        ss.append(np.polyld(coeffs[i],variable ="(x-"+str(xs[i])+")")) # shift
41      else:
42        coeffs.append([ys[i],bs[i],cs[i],ds[i]]) # npp.Polynomial: high-to-low
43        ss.append(npp.Polynomial(coeffs[i]))
```

where a_i = *ys[i]* and b_i, c_i, d_i wrt to cubic spline $S_i(x)$ as below.

$$S_i(x) = a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \text{ on } [x_i, x_{i+1}] \text{ for } i = 1, \ldots, n-1.$$

why are the coefficients (ie, variable "coeffs") reversed between the calls to numpy.polyld() and numpy.polynomial.Polynomial()?

q5. (5 pts **MATH 685 ONLY**) generalized minimum residual (GMRES) is an iterative method. given inital guess x_0, what does this method iterate? ie, *how* does this method iterate?

q6. (5 pts **MATH 685 ONLY**) what is λ to the levenberg-marquardt method?