

FINAL project part A: linear interpolation

Play by the rules:

1. The use of any form of cycles or symbolic toolbox functions in you implementation is prohibited. Using the vector features of MATLAB is mandatory.
2. It is prohibited to use any toolbox function when implementing `linterpCoef()`. It is mandatory to implement it using the equations derived in the lecture.

Exercise 1: Preparation

Read the documentation for the following MATLAB functions: `rand`, `randn`, `randi`, `mean`, `var`, `cov`, `polyfit`. Read the short help provides by the `>>help function` command, and also the more detailed help provided by the `>>doc function` command. Except where explicitly stated otherwise, implement all code into a single .m file using cell mode (use the cell separator: `%%`). For further examples look at the MATLAB linear regression documentation online (in english): https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html

Implement exercise 2 into `linterpCoef.m` and `linterp.m`, use `lin_interpolation.m` as a unit test, do not modify this file. Implement exercises 3,4,5 directly into `exercise345.m`.

Exercise 2: Linear interpolation

Into a separate .m file implement a function `linterpCoef()` that takes two points p_1 and p_2 in 2D plane as input parameters and returns the coefficients a, b of a linear function $f(x) = ax + b$ that crosses these points. The points are defined as vectors of two values: $p_i = [x_i, y_i]$. Into another .m file implement a second function with a prototype given: `[X, Y] = linterp(a, b, X1, n)` that calculates the values of the linear interpolation function $f(x) = ax + b$ for an interval of x values given by $X_1 = [x_{\min}, x_{\max}]$ where $n \geq 1$ is the number of intervals into which the interval X_1 will be divided. The function should return two vectors: the interpolated coordinates x and corresponding values of $f(x)$ at these points. The length of the resulting vectors should be $n+1$ and the x values should include the boundary values x_{\min} and x_{\max} .

Exercise 3: Generate pseudo-random matrices

Implement the public function: `function [Rx] = random(Lim, Si, flag)`, to generate matrices of pseudo-random numbers with a uniform or normal distribution. The first parameter `Lim = [a b]`, the second means size of the resulting matrix – vector `Si = [M N]`. The last parameter – the `flag` string is optional, and if it is specified, it can only take on the values "uniform" or "normal". For the value "normal" then the function generates values with a normal distribution, and the parameters `[a b]` actually define the desired mean and variance, otherwise it generates values with a uniform distribution from the interval `[a b]`. The implicit value of the `flag` parameter should be defined inside the function as "uniform".

Using this function in a main script, then generate a matrix R_u of pseudo-random floating point values from a uniform distribution, a matrix R_i of pseudo-random integer values from a uniform distribution and a matrix R_n of pseudo-random values from a normal distribution. Let the size of all matrices be $M \times N$. Let the expected value of R_n be $m = 8$ and variance $v = 3$. Let the values of R_u and R_i be from the interval $a, b = < 4, 9 >$.

Calculate the sample mean and sample variance of all pseudo-random matrices:

- a) for each column (The results will be row vectors of sample averages and variances.)
- b) for the whole matrix (The results will be two scalar numbers per each matrix: sample mean and sample variance.)

Note: You don't have to implement the formulas defined in the section below, just use the MATLAB built-in functions.

Exercise 4: Calculate basic statistics

Generate a pseudo random column vector V of varying length N (using the already prepared `for` cycle containing also a print statement) and mean and variance given: $m = 10$, $v = 3$. For each vector length N calculate sample mean m_s and variance v_s and for each calculated m_s and v_s calculate their relative errors in percent (That means error with respect to the given values $m = 10$, $v = 3$). Also print these values. The errors should converge to zero with large N .

Exercise 5: Generate noisy data

Implement a public function: `function [x,y] = noisyLinear(alpha, beta, X1, step, v)`, which will generate the values of the linear function $f(x) = \alpha x + \beta$ for the values of the vector x from the interval $X_1 = [X_{\min}, X_{\max}]$ with step `step`. At the same time, pseudorandom noise with a normal probability distribution with variance v is added to the values of the linear function. Let the mean value of the noise vector be set to zero.

Using this function, then generate the values of the linear function $f(x) = ax + b$ for the values of the vector x from the interval $< -4, 10 >$ with a step of 0.5. Create two noisy vectors Y_1 and Y_2 simulating measurements of real quantities: one with variance $v_1 = 0.1$, the other with variance $v_2 = 2$. Plot all three waveforms in one window side by side: first the pure linear function $f(x)$ using the `plot()` function, then a slightly noisy course of Y_1 observations using the `scatter()` function, and finally a very noisy course of Y_2 (again using `scatter()`). Use `subplot()` to separate the plots.

FINAL projekt časť A: lineárna interpolácia

Pravidlá:

1. Pri implementácii úloh je zakázané používať akékoľvek cykly, alebo funkcie symbolického toolboxu. Využitie vektorových operácií systému MATLAB je povinné.
2. Pri implementácii funkcie `linterpCoef()` je zakázané používať akékoľvek funkcie toolboxov, implementovať je ju nutné pomocou vzorcov odvodených na prednáške.

Cvičenie 1: Príprava

Prečítajte si dokumentáciu knasledujúcim funkciám MATLABu: `rand`, `randn`, `randi`, `mean`, `var`, `cov`, `polyfit`. Prečítajte si nie len krátku verziu dokumentácie poskytovanú pomocou príkazu: `>>help function`, ale aj detailnejšiu dokumentáciu, ktorú zobrazuje príkaz: `>>doc function`. Okrem úloh, kde je explicitne uvedený iný postup, implementujte všetky úlohy do jedného spoločného .m súboru, pričom použite bunkový režim (oddelenie pomocou: `%`) pre oddelenie implementácie jednotlivých úloh. Prečítajte si podrobnú dokumentáciu k implementácii lineárnej regresie v angličtine: https://www.mathworks.com/help/matlab/data_analysis/linear-regression.html.

Implementujte cvičenie 2 so súborov `linterpCoef.m` and `linterp.m`, použite `lin_interpolation.m` ako unit test - pre kontrolu správnosti (tento súbor nemodifikujte). Implementujte cvičenia 3,4,5 priamo do súboru `exercise345.m`.

Cvičenie 2: Lineárna interpolácia

Do samostatného .m súboru implementujte funkciu `linterpCoef()`, ktorá má vstupné parametre dva body v 2D rovine: $p1$ a $p2$, a ktorej výstupné parametre sú koeficienty a , b lineárnej funkcie $f(x) = ax + b$, ktorá prechádza týmito bodmi. Body sú definované ako vektory $p_i = [x_i, y_i]$. Do ďalšieho .m súboru implementujte druhú funkciu s prototypom `[X,Y] = linterp(a, b, X1, n)`, ktorá vypočíta hodnoty lineárnej interpolácie pre zadaný interval x hodnôt $X_1 = [x_{\min}, x_{\max}]$ kde $n \geq 1$ je počet podintervalov rovnakej dĺžky, na ktoré sa má rozdeliť interval X_1 . Funkcia má vracať dve hodnoty: vektor interpolovaných x-súradníc a vektor korešpondujúcich hodnôt funkcie $f(x)$. Dĺžka oboch týchto vektorov má byť $n+1$ a krajné hodnoty výstupného vektora X majú byť x_{\min} a x_{\max} .

Cvičenie 3: Generovanie pseudo-náhodných matíc

Implementujte public funkciu: `function [Rx] = random(Lim, Si, flag)`, pre generovanie matíc pseudo-náhodných čísel s rovnomerným, alebo normálnym rozdelením. Prvý parameter definuje rozsah generovaných hodnôt $Lim = [a, b]$, druhý parameter rozmery generovanej matice $Si = [M, N]$. Posledný parameter – reťazec `flag` je nepovinný, a ak je špecifikovaný, môže nadobúdať len hodnoty "uniform" alebo "normal". Pre hodnotu "normal" potom funkcia generuje hodnoty s normálnym rozdelením, a parametre $[a, b]$ v skutočnosti definujú požadovanú strednú hodnotu a varianciu, inak generuje hodnoty s rovnomerným rozdelením z intervalu $[a, b]$. Implicitná hodnota paramtera `flag` nech je definovaná vo vnútri funkcie ako "uniform".

V hlavnom programe potom vygenerujte maticu R_n pseudonáhodných hodnôt (desatinných čísel typu double) s rovnomerným rozdelením pravdepodobnosti. Ďalej vygenerujte maticu pseudonáhodných celých čísel R_i s rovnomerným rozdelením pravdepodobnosti a maticu pseudonáhodných desatinných čísel R_n s normálnym rozdelením. Nech rozmery všetkých matíc $M \times N$ sú určené preddefinovanými premennými M a N . Očakávaná stredná hodnota pre maticu R_n nech je daná $m = 8$ a variancia $v = 3$. Zabezpečte, aby všetky prvky vygenerovaných matíc R_n a R_i ležali v intervale: $\langle a, b \rangle = \langle 4, 9 \rangle$.

Vypočítajte výberový priemer a varianciu pre všetky vygenerované pseudo-náhodné matice:

- a) pre každý stĺpec matice zvlášť (Výsledkom bude riadkové vektory výberových priemerov a variancií.)
- b) naraz pre celú maticu (Výsledkom budú dve skalárne čísla pre každú maticu: výberový priemer a variancia.)

Poznámka: Vzorce uvedené ďalej v samostatnej sekcii nemusíte implementovať, stačí zavolať už existujúce funkcie MATLABu.

Cvičenie 4: Vypočítajte základné štatistiky

Vygenerujte pseudonáhodný stĺpcový vektor V , ktorého dĺžku danú premennou N budete meniť v cykle (v programe je už predpripravený cyklus `for` aj s príkazom pre výpis). Nech očakávané hodnoty strednej hodnoty a variancie sú konštantné $m = 10$, $v = 3$. Pre každú dĺžku N vektora vypočítajte výberový priemer m_s a výberovú varianciu v_s a pre každú hodnoha $=tu$ m_s a v_s vypočítajte ich relatívnu chybu v percentách (Teda chybu voči očakávaným hodnotám $m = 10$ a $v = 3$). Vypíšte tieto hodnoty. Hodnoty chýb by mali klesať spolu so zvyšujúcou sa hodnotou N .

Cvičenie 5: Vygenerujte zašumené dáta

Implementujte public funkciu : `function [x,y] = noisyLinear(alpha, beta, Xl, step, v)`, ktorá vygeneruje hodnoty lineárnej funkcie $f(x) = \alpha x + \beta$ pre hodnoty vektora x z intervalu $Xl = [Xmin, Xmax]$ s krokom `step`. K hodnotám lineárnej funkcie zároveň pripočíta pseudonáhodný šum s normálnym rozdelením pravdepodobnosti s varianciou v . Stredná hodnota šumového vektora nech je nastavená na nulu.

Pomocou tejto funkcie potom vygenerujte hodnoty lineárnej funkcie $f(x) = ax + b$ pre hodnoty vektora x z intervalu $<-4,10>$ s krokom 0.5. Vytvorte dva zašumené vektory Y_1 a Y_2 simulujúce merania skutočných veličín: jeden s varianciou $v_1 = 0.1$, druhý s varianciou $v_2 = 2$. Vykreslite všetky tri priebehy do jedného okna vedľa seba: najprv čistú lineárnu funkciu $f(x)$ pomocou funkcie `plot()`, potom mierne zašumený priebeh pozorovaní Y_1 pomocou funkcie `scatter()`, a nakoniec veľmi zašumený priebeh Y_2 (opäť pomocou `scatter()`). Použite `subplot()` pre oddelenie grafov.

Equations / Rovnice**Basic statistics / Základné štatistiky**

Sample mean, výberový priemer	$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$	(1)
-------------------------------	--	-----

Sample variance, výberová variancia	$s_x^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$	(2)
-------------------------------------	--	-----

Sample covariance, výberová kovariancia	$s_{x,y} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$	(3)
---	---	-----

Calculating simple linear regression coefficients**Výpočet koeficientov jednoduchej lineárnej regresie**

$\beta = \frac{s_{x,y}}{s_x^2}$	(4)
---------------------------------	-----

$\alpha = \bar{y} - \beta \bar{x}$	(5)
------------------------------------	-----

Calculating residuals and coefficient of determination**Výpočet reziduí a koeficientu determinácie**

$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$	(6)
---	-----

$r_i = y_i - \hat{y}_i$	(7)
-------------------------	-----