ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Models, Algorithms and Data (MAD): Introduction to Computing

Spring semester 2021

Prof. Dr. Jens Honore Walther
Dr. Georgios Arampatzis
ETH Zentrum, CLT
CH-8092 Zürich

Exam

Issued: Tuesday, 24.08.2021

Exam directives. In order to pass the exam, the following requirements have to be met:

- Read carefully the first two pages of the exam. Write your name and Legi-ID where requested. Before handing in the exam, **PUT YOUR SIGNATURE ON PAGE 2**.
- Clear your desk (no cell phones, cameras, etc.): on your desk you should have your Legi, your pen, paper and your summary.
- Calculators are not permitted.
- If necessary the teaching assistants will give you additional paper sheets. On the top-right corner of every page write your complete name and Legi-ID.
- The personal summary consists of no more than 4 pages (2 sheets). The personal summary can be handwritten or machine-typed. In case it is machine-typed, the text has to be single-spaced and the font size has to be at least 8 pts.
- You can answer in English or in German; the answers should be handwritten and clearly readable, written in blue or black - do NOT write anything in red or green. Only one answer per question is accepted. Invalid answers should be clearly crossed out.
- If something is disturbing you during the exam, or it is preventing you from peacefully solving the exam, please report it immediately to an assistant. Later notifications will not be accepted.
- You must hand in: the exam cover, the sheets with the exam questions and your solutions. The exam cannot be accepted if the cover sheet or the question sheets are not handed back.
- **You need 100 points out of 120 for a 6.**

Family Name:	
Name:	
Legi-ID:	

Question	Maximum score	Score	TA 1	TA 2
1 - 7	30			
8	10			
9	10			
10	25			
11	10			
12	15			
13	5			
14	15			
Total	120			

With your signature you confirm that you have read the exam directives; you solved the exam without any unauthorized help and you wrote your answers following the outlined directives.

Signature:

GOOD LUCK!

Multiple Choice Questions [30 Points]

In the following Multiple Choice Questions, you are asked to tick the correct answers using a 'X' in the corresponding box. In total there are **20 correct answers** in Questions 1-7. Each correctly selected answer is rewarded with 1.5 points. If a correct answer is not selected it is rewarded 0 points. **If you select a wrong answer, 1.5 points are deducted**. The minimal number of points for your answers to all multiple choice questions is 0 out of 30 points.

Question 1: Linear Least Squares

Assume that you did an experiment, and you plotted the resulting data $\{(x_i, y_i)\}_{i=1}^{N=100}$ in Figure 1.

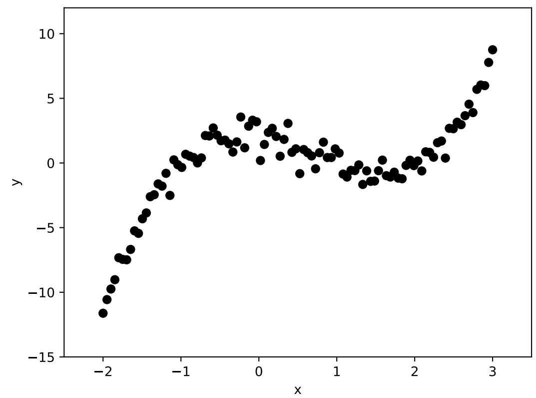

Figure 1: Data from an experiment.

a) You want to find the function $f(x)$ that best approximates the data $y_i \approx f(x_i)$ using a linear least squares fit. This requires

- ☐ selecting the measure of "best".
- ☐ selecting a set of basis functions.
- ☐ initializing the weights.
- ☐ knowing the physical law underlying the data.

b) The data plotted in Figure 1 suggests that we should use

- ☐ a polynomial basis $\varphi_k(x) = x^k$
- ☐ an exponential basis $\varphi_0(x) = \exp(kx)$
- ☐ a trigonometric basis $\varphi_k(x) = \cos(kx)$

for $k = 0, 1, 2, 3$

c) The inversion of the matrix $A^\top A$ in the normal equation $A^\top A\boldsymbol{w} = A^\top \boldsymbol{y}$

- ☐ is impossible, because it has a very high condition number.
- ☐ can always be done analytically.
- ☐ is possible if the chosen basis functions are linearly independent.

Question 2: Nonlinear systems

We want to solve a general non-linear equation $f(x^\star) = 0$ for the unknown root $x^\star$.

a) The order of convergence can be computed from iterations $x_0, \ldots, x_k$ of the algorithm by

- ☐ taking two subsequent errors $e_{k-1} = x_{k-1} - x^\star$ and $e_k = x_k - x^\star$ and computing

$$r = \frac{\log |e_k|}{\log |e_{k-1}|} .$$

- ☐ cannot be computed as it is only defined in the limit $k \to \infty$.
- ☐ computing the sequence of errors $e_k = x_k - x^\star$ and estimating the order r as

$$r \approx \frac{\log \left|\frac{e_{k+2}}{e_{k+1}}\right|}{\log \left|\frac{e_{k+1}}{e_k}\right|} .$$

b) The bisection method is guaranteed to converge. This statement is

- ☐ true
- ☐ false

Question 3: Interpolation with Lagrange Polynomials and Cubic Splines

a) The following subquestions consider the interpolation of data from a 2π periodic function, which is a linear combination of $\cos(x)$ and $\sin(x)$.

1. We interpolate the function with Cubic Splines interpolation, using 3, 4 and 7 sampling points in $[0, 2\pi]$. The resulting interpolating polynomials are shown in Figure 2. Select the correct combination of letters, indicating the corresponding line in the graph, and the number of points that were used for the interpolation.
   - ☐ (a): 4, (b): 7, (c): 3
   - ☐ (a): 3, (b): 4, (c): 7
   - ☐ (a): 7, (b): 3, (c): 4

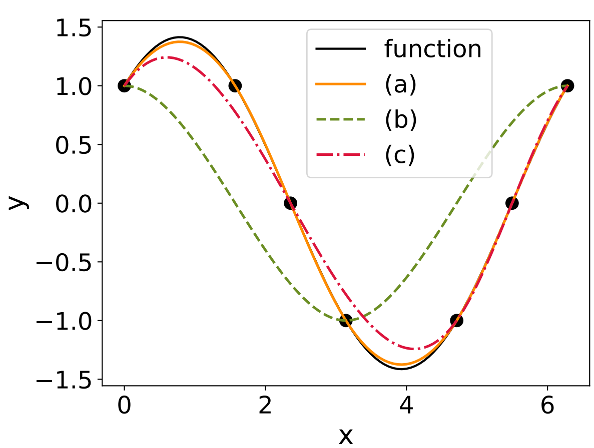

Figure 2: Interpolation of a periodic function.

2. We now consider data with a Gaussian random error term, i.e. $y_i = f(x_i) + \varepsilon$, where $i = 1, \ldots, N$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. In Figure 3, select the correct combination of letters indicating the line in the graph, and interpolating method with N interpolation points.

   - ☐ (a): lagrange interpolation $N = 20$, (b): cubic splines $N = 20$, (c): lagrange interpolation $N = 7$.
   - ☐ (a): cubic splines $N = 7$, (b): lagrange interpolation $N = 20$, (c): cubic splines $N = 7$.
   - ☐ (a): cubic splines $N = 20$, (b): lagrange interpolation $N = 20$, (c): lagrange interpolation $N = 7$.

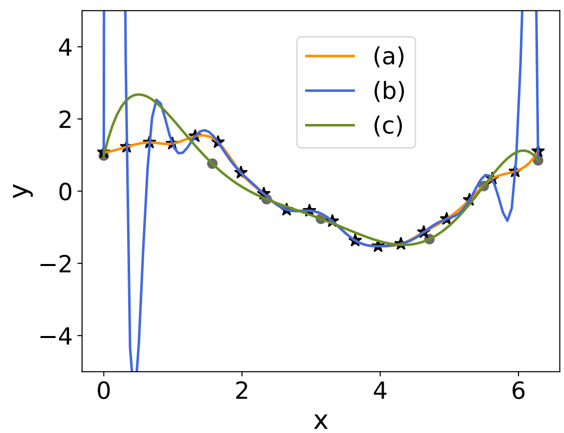

Figure 3: Interpolation of a periodic function with noisy data.

b) Which of the following statement(s) for Lagrange and cubic spline interpolation is/are correct?

- ☐ The Lagrange interpolating polynomial through 7 sample points from $f(x) = \frac{1}{2}x^4 + 2x + 3$ without addition of noise will be the original function $f(x)$.
- ☐ If I want to interpolate through the data points: $x_1 = (1, 3)$, $x_2 = (2, 5)$ and $x_3 = (3, 7)$ with Lagrange interpolation, the second Lagrange polynomial will correspond to: $l_2(x) = -(x^3 - 4x + 3)$.
- ☐ For a dataset of 10 non-overlapping datapoints, the Lagrange interpolating function $p(x)$ will have 10 roots.
- ☐ For any boundary condition, we can use the Thomas Algorithm (TDMA) to solve the system of the unknown second derivatives on the data nodes.

Question 4: Numerical Quadrature

In the next multiple choice questions, choose the correct answer(s).

a) The two-segment trapezoidal rule is able to exactly integrate at most polynomials of order:

- ☐ one
- ☐ two
- ☐ three
- ☐ four

b) For a second order (quadratic) polynomial, the two-point Gauss quadrature rule will give the same result as the integration rule:

- ☐ Rectangle Rule
- ☐ Midpoint Rule
- ☐ Simpson's Rule
- ☐ Trapezoidal Rule

c) Deriving a three-point Gauss quadrature requires computing ____ unknowns:

- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 6

Question 5: Romberg extrapolation and adaptive integration

a) The approximation error of an integral with n intervals of width h is given by $I_0^n = I - c_1h^3 - c_2h^6 - c_3h^9 + \dots$. Using Richardson extrapolation and the integral evaluated at twice the intervals I_0^{2n} we can eliminate the $\mathcal{O}(h^3)$ terms. Select the correct expression(s) for the error of the resulting approximation I_1^n.

- ☐ $I_1^n = I + \frac{1}{4}c_2h^6 + \frac{9}{16}c_3h^9 ...$
- ☐ $I_1^n = I + \frac{1}{4}c_2h^6 + \frac{9}{32}c_3h^9 ...$
- ☐ $I_1^n = I + \frac{1}{8}c_2h^6 + \frac{9}{64}c_3h^9 ...$
- ☐ $I_1^n = I + \frac{1}{8}c_2h^6 + \frac{9}{128}c_3h^9 ...$

b) Mark the correct statement(s) regarding adaptive integration:

- ☐ Adaptive integration can lead to an equispaced (uniform) subdivision of the whole interval.
- ☐ Adaptive integration can lead to a non-uniform subdivision of the whole interval.
- ☐ In areas of large function values we expect a more fine grained subdivision of the integration interval than in areas of small function values.
- ☐ Using adaptive integration, the desired accuracy is always smaller than the integration error.

Question 6: Probability Theory and Monte Carlo Integration

In the next multiple choice questions, choose the correct answer(s).

a) Mark the correct statement(s) regarding probability density functions.

- ☐ A probability density function is always non-negative.
- ☐ The standard deviation of a random variable is always smaller than the expectation value.
- ☐ Continuous probability density functions integrate up to one.
- ☐ The cumulative distribution function F is strictly monotonically increasing (i.e. $F(a) > F(b)$ if $a > b$).

b) Assume a uniform distribution $\mathcal{U}([0,1])$ which is defined in an interval $[0,1]$. The variance of a random variable sampled from this probability distribution is

- ☐ 1
- ☐ $\frac{1}{3}$
- ☐ $\frac{1}{6}$
- ☐ $\frac{1}{12}$

c) The Monte Carlo method for integration has higher order than the Simpson's rule for any dimension higher than

- ☐ 1
- ☐ 2
- ☐ 4
- ☐ 8

d) In order to reduce the error ϵ_M of the Monte Carlo integral estimator by a factor of 10, how many more samples are required?

- ☐ $\sqrt{10}$
- ☐ 10
- ☐ 100
- ☐ 1000

e) Using the inverse transform sampling method, we wish to generate a random sample X sampled from the Bernoulli distribution with $P(X=1)=\frac{2}{3}$ and $P(X=0)=\frac{1}{3}$. First, we generated a sample from the uniform distribution $u=0.6$. The corresponding sample X evaluates as

- ☐ 0
- ☐ 0.6
- ☐ 2/3
- ☐ 1

Numerical Problems [70 Points]

The following exercises involve practical application of the lecture concepts. **Please use one separate sheet of paper for each of the questions.**

Question 7: Linear Least Squares [10 Points]

Assume you are given the following data with time t and distance d for a car passing by:

$t[s]$	1	2	3	4	5
$d[m]$	4.5	9	10.5	16	17

The distance for a car with velocity v starting from an initial position d_0 takes the form

$$d(t; d_0, v) = d_0 + vt\,. \tag{1}$$

Perform a linear-least squares fit to determine the unknown coefficients $d_0^\star, v^\star$.

Question 8: Newtons Method [10 Points]

You are given the function

$$f(x) = 3x^2 - 2\sin(x)\,, \tag{2}$$

for which you want to find the root $x^\star$, i.e.

$$f(x^\star) = 0\,. \tag{3}$$

a) Write down the Newton iteration and compute the derivative $f'(x)$ of the function $f(x)$.

b) Given the initial guess $x_0 = \pi/2$, perform an iteration of Newtons method.

c) Assume you are given the following iterations of a non-linear solver:

x_2	x_3	x_4	$\cdots$	$x^\star$
0.7312	0.6386	0.6245	$\cdots$	0.6242

Provide an crude estimate of the order of convergence based on these iterations and make an educated guess what method produced this sequence.

Hint: Take into account the order of magnitude, i.e. $0.03 \approx 10^{-2}$.

Question 9: Cubic Splines [25 Points]

We are trying to fit the simple cubic function $f(x) = x^3$ with cubic spline interpolation, using three datapoints sampled from the function: (x_i, y_i), with $i = 0, 1, 2$: $[(0,0), (1,1), (2,8)]$.

a) We first try to interpolate through the three datapoints with cubic splines with natural boundary conditions. Write down the matrix system you need to solve for the second derivatives on the data nodes for natural boundary conditions. Derive the coefficients of the matrix system and solve the system. Finally, write down the piecewise cubic spline for the intervals defined by the three points. (10 points))
Note: You can use the formula: $(a-b)^3 = a^3 - 3a^2b + 3ab^2 - b^3$.

b) The resulting piecewise cubic polynomials are depicted in Figure 4 below, along with the actual function curve. We notice that the interpolating function is not the same as the original function, although we are fitting a piecewise cubic function! Why do you think this discrepancy between the two curves occurs? Can you propose a solution to this problem? (5 points)

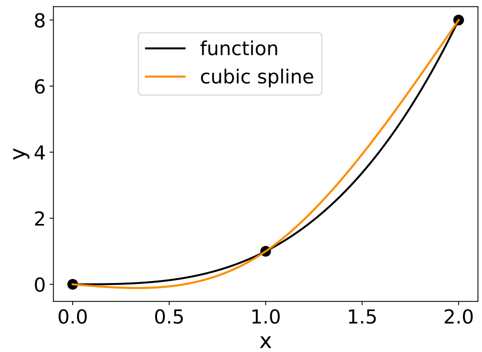

Figure 4:

c) You then consider applying the parabolic runout boundary conditions for splines to your problem. Remember that this means that $f_0'' = f_1''$ and $f_{N-1}'' = f_N''$ for a system of N equations. Rewrite and solve the matrix system for the second derivatives on the data nodes for the parabolic runout boundary condition. What can you say regarding the resulting splines, without deriving the final formulas? How do you think that this boundary condition is going to affect the interpolating spline? (10 points)

Question 10: Numerical Quadrature [10 points]

You are asked to estimate the water flowrate in a pipe of radius $R = 2$ m. The flowrate Q is defined as the integral of the velocity along the pipe circumference over the radial direction, where $r = 0$ is the center of the pipe and $r = R = 2$ is the end point, i.e. the radius of the pipe,

$$Q = \int_0^2 2\pi r \nu \mathrm{d}r \ . \tag{4}$$

You do not know how the velocity ν varies along the radial direction and due to technical reasons you are asked to use exactly two measurement points of the velocity along the radial direction of the pipe.

a) Which method of numerical integration, that was taught in the lecture, do you choose in order to most accurately predict the water flow rate, given that you only are allowed to use two probing points?

b) Compute the two probing points you need to design your measuring system, i.e. r_1, r_2 with $r_1 < r_2$.
Note: you may use the fact that $\frac{1}{\sqrt{3}} = 0.58$

c) You end up manufacturing your measuring system and you probe the velocity at the afore-calculated locations $\nu_1 = \nu(r_1) = 1$ m/s and $\nu_2 = \nu(r_2) = 2$ m/s. Compute the water flowrate for your measurement conditions.

Question 11: Backpropagation and neural networks [15 points]

A neural network architecture is assembled using a 3-neuron input layer, a single-neuron hidden layer, and a 3-neuron output layer. It employs identity and ReLU activation functions on the hidden and output layers (respectively), but does not make use of any biases.
The ReLU function is defined as:

$$\phi(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

a) Draw the network, and label the input layer $\boldsymbol{x}$, hidden layer $\boldsymbol{h}$, and output layer $\boldsymbol{y}$.

b) Perform a forward pass using a sample $\boldsymbol{x}^{(1)} = [4, 2, 1]$, and compute the reconstruction loss $E^{(1)} = \|\boldsymbol{x}^{(1)} - \boldsymbol{y}^{(1)}\|_2^2$. The current weights of the hidden and output layers are given as follows:

$$\mathbf{W}^h = \begin{bmatrix} 3 & -2 & -4 \end{bmatrix}, \mathbf{W}^y = \begin{bmatrix} 2 \\ 1 \\ -3 \end{bmatrix}$$

c) Perform the weight update rule for $W_2^{y,new}$ given the current weight using a gradient descent step with $\eta = 0.10$.

d) A new fully-connected linear layer is added right after the input layer (with the same number of nodes as the input). Comment on whether we should expect enhanced expressiveness of the network.

e) We take the network from (d) and decide to add biases in every layer, count the total number of parameters of the resulting network.

f) Taking the original network from (a), do you think the architecture is well-suited for an autoencoder?

Pseudocode [20 Points]

The last part of the exam consists of a practical implementation of the learned concepts in a pseudocode. **Please use one separate sheet of paper for each of the questions.**

Question 12: Pseudocode for Romberg integration [5 Points]

Below in Algorithm 1 you find a pseudocode for the Romberg integration. Each of the three main blocks contain one error. Identify the three errors and correct them.

Algorithm 1 Romberg integration

Input:

function $f(x)$
interval boundaries a, b
number of iterations K

Output:

$I_K^1 = \texttt{integral}[K, 0]$ approximation to the integral $\int_a^b f(x)\,\mathrm{d}x$

Steps:

```
maxNumIntervals ← 2^K

// Precompute and store function evaluations (Block 1)
hmin ← 1/maxNumIntervals
for i ← 0, ..., maxNumIntervals do
    fvalues[i] ← f(a + i * hmin)
end for

// Compute level 0 integrals (Block 2)
for r ← 0, ..., K do // refinement
    numIntervals ← 2^r
    step ← 2^(K−r) // step between two function evaluations for this refinement
    result ← 0
    for i ← step, 2 * step, 4 * step, 8 * step ..., maxNumIntervals − step do
        result ← result + fvalues[i]
    end for
    // composite trapezoidal rule:
    integral[0, r] ← 0.5 (b−a)/numIntervals (fvalues[0] + fvalues[maxNumIntervals]
                                              +2 * result)
end for

//Advance to higher precision according to Romberg (Block 3)
for l ← 1, ..., K do //level
    for r ← 0, ..., K − l do // refinement
        integral[r, l] ← (4^l * integral[l−1, r+1] − integral[l−1, r]) / (4^l − 1)
    end for
end for
```

Question 13: PCA for Dimensionality Reduction [15 Points]

Given a sample dataset $X = [x_1, x_2, ..., x_N]$ with $x_i \in \mathbb{R}^D$, write a pseudocode showing all the steps needed to perform a dimensionality reduction of the dataset using PCA. Your goal is to keep at least 90% of the original variance. Make sure you specify the dimensions of each matrix or vector in your pseudocode, and clearly indicate the input, output and steps of the algorithm. To assist you in your work, you are provided with the following helper functions:

- `computeMean(X, shape, dim)`: Computes the mean of matrix `X` of specified `shape` over the dimensions `dim`. Specify the last two arguments as a tuple e.g. `shape`=(5, 4, 2). `dim`=2.
- `transpose(X)`: Returns the transpose of an input matrix `X`.
- `L, V = eigenDec(X)`: Computes the eigendecomposition of an input matrix `X`, returning a vector of eigenvalues `L`, as well as a matrix `V` containing the associated eigenvectors (in no predefined order).
- `newVec, shiftVec = sortVec(X, ord)`: Sorts a vector in ascending or descending order. Specify `ord=asc` or `ord=desc` if you wish to sort it in ascending or descending order (respectively). An indexation vector of the shifting (`shiftVec`) is also given. For example, after sorting the initial vector `[5, 6, 8, 7]` in descending order, the vector `newVec` become `[8, 7, 6, 5]`, and the corresponding the shifting indexation vector is `[2, 3, 1, 0]` - index 2 is placed first because it contains the largest value in the input vector.
- `vec_sum(a)`: Returns the sum of the elements of a vector a. If a scalar is used as input, the output will be that same scalar.