ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

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

Spring semester 2018

Prof. P. Koumoutsakos, Prof. J. H. Walther
ETH Zentrum, CLT
CH-8092 Zürich

Solution

Issued: Saturday, 10.08.2019

Short Questions [± 1 Point per Subquestion, min 0 per Question, max 24]

Question 1: Multi-dimensional nonlinear solver

a) Bisection, Newton and Secant methods all work well for one dimensional nonlinear problems. Which can be extended directly to multi-dimensional nonlinear problems?

- ☐ Bisection and Newton
- ☐ Bisection and Secant
- ■ Newton and Secant
- ☐ Newton

b) What is the right ordering in terms of order of convergence (from fastest to slowest)

- ☐ Newton, Bisection, and Secant
- ☐ Bisection, Secant, and Newton
- ■ Newton, Secant, and Bisection
- ☐ Secant, Newton, and Bisection

c) Which of the following statements is true?

- ☐ If the initial guess is around the true solution, the Newton method has always quadratic convergence rate.
- ■ The sufficient conditions for the minimum of $E(\vec{x})$ are: $\nabla E = 0$ and Hessian matrix of E is positive semidefinite at x^*.
- ☐ Newton method is not proper for a least squares problem.
- ☐ The the problem of minimization for $E(\vec{x})$ is exactly the same as solving a non-linear equation $\nabla E(\vec{x}) = 0$.

Question 2: Lagrange interpolation

a) The Lagrange basis functions are equal to zero for all $x_{k\neq j}$ and equal to one for x_j. Therefore we can also write the interpolating function as $f(x) = \sum_{k=1}^{N} y_k \delta_{kj}$, where δ_{kj} is the Kronecker delta function:

☐ True

■ False

b) Assume we are able to sample points from a quadratic function. How many points are required such that Lagrange interpolation exactly resembles the quadratic function that we sampled from?

☐ 2

■ 3

☐ 4

☐ 6

c) Assume you have a dataset consisting of 5 points. How many roots does each single term of the interpolating function have, ie. $l_i(x) = 0$?

☐ 3

■ 4

☐ 5

☐ Depends on the datapoints

d) You used Lagrange interpolation to interpolate a set of data points. However, you lose the original data set and only have an incomplete graph, figure 1. What might the data points have been?

☐ $\{(5, 1), (15, 1)\}$

☐ $\{(5, 2), (20, 10)\}$

☐ $\{(15, 9), (20, 10)\}$

☐ $\{(3.1, 1.1), (5, 2), (20, 10)\}$

■ $\{(5, 2), (15, 9), (20, 10)\}$

☐ Cannot be said from the given information

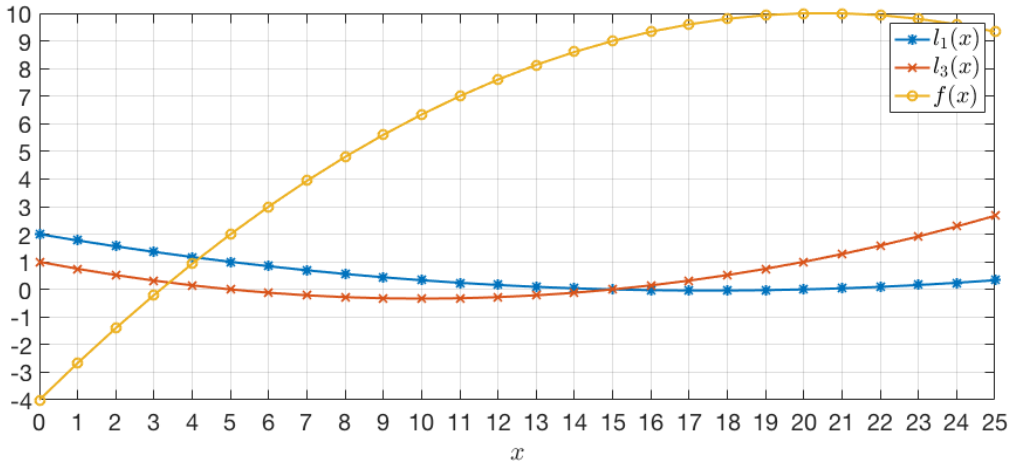


Figure 1: Lagrange interpolation $f(x)$ and two basis function $l_1(x)$ and $l_3(x)$.

e) You have an arbitrary dataset $(x_1, y_1), ..., (x_N, y_N)$ and decide to use Lagrangian interpolation:

$$f(x) = \sum_{k=1}^{N} w_k l_k(x)$$

Here $l_k(x)$ denote to the Lagrange functions introduced in the lecture. We want to perform an optimization for the parameters w_k using least squares:

$$w_1, ..., w_N = \arg \min_{w_1,...,w_N} \sum_{k=1}^{N} (w_k l_k(x_k) - y_k)^2$$

Once you have obtained the weights and plugged them into your model, your model will be

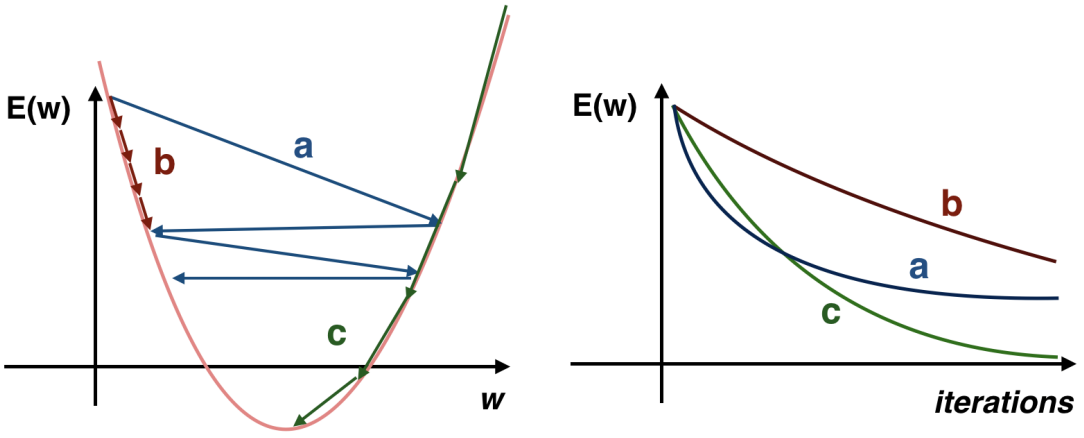
- ■ ... the same as when you would have used Lagrange interpolation
- ☐ ... slightly different than the Lagrange interpolation due to the square term in the cost function
- ☐ ... completely different since the least square approach is not comparable with Lagrange interpolation

Question 3: Artificial Neural Network Training

Artificial neural networks are trained by stochastic gradient descent. In it's most basic form, this method has a single parameter, the learning rate η.

a) Each training curve on the left graph corresponds to a curve on the right graph. Each pair of curves correspond to a different learning rate. Select the correct combination of curves:

- ☐ (a,1), (b,3) (c,2)
- ■ (a,1), (b,2) (c,3)
- ☐ (a,2), (b,1) (c,3)
- ☐ (a,2), (b,3) (c,1)



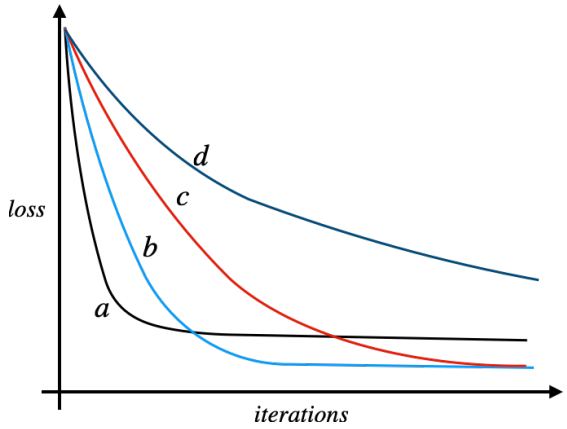
b) Each curve on the graph correspond to the training history of a network with a different learning rate. Select the curve corresponding to the optimal learning rate:

☐ a

■ b

☐ c

☐ d



c) Only one of this statement about gradient descent (GD) is true. Select the true statement:

- ☐ GD is guaranteed to converge to the global optimum.
- ☐ A larger learning rate guarantees that the model will converge to an optimal solutions in fewer iterations.
- ■ GD can converge to a local optimum.
- ☐ Multiple initialization of a same neural network trained on the same data will converge to the same optimum.

Question 4: Richardson Extrapolation

a) ☐ True

■ False

The concept of Richardson extrapolation is a simple and elegant way to extend the accuracy by which we compute numerically any quantity.

b) ■ $a = 1$

☐ $a = \frac{1}{2}$

☐ $a = \frac{1}{3}$

Here we use that if $L = L(h) + k_1 h + k_2 h^2 + ...$ then it holds:

$$L_n(h) = \frac{1}{2^n - 1}(2^n L_{n-1}(\frac{h}{2}) - L_{n-1}(h)) \tag{1}$$

For $n = 1$ one gets:

$$L_1(h) = (2L_0(\frac{h}{2}) - L_0(h)) = L_0(\frac{h}{2}) + 1 \cdot (L_0(\frac{h}{2}) - L_0(h)) \tag{2}$$

$\rightarrow a = 1$

c) ☐ $a = 1$

☐ $a = \frac{1}{3}$

■ $a = \frac{1}{15}$

Now we have $L = L(h) + k_1 h^2 + k_2 h^4 + O(h^6)$. To perform extrapolation, we want to take a linear combination of L(h) and L(h/2) so that its error term is of order 4:

$$L = L(h) + k_2 h^2 + k_4 h^4 + O(h^6) \tag{3}$$

$$L = L(\frac{h}{2}) + k_2 (\frac{h}{2})^2 + k_4 (\frac{h}{2})^4 + O(h^6) \tag{4}$$

Multiply the second equation above by 4 and subtract the first equation from it, finally divide both sides by 3 to obtain:

$$L_1(h) = \frac{4}{3}L(\frac{h}{2}) - \frac{1}{3}L(h) - k_4\frac{1}{4}h^4 + O(h^6) \tag{5}$$

Now we have to apply second time the Richardson's extrapolation, i.e. so that its error term is of order 6:

$$L = L_1(h) = \frac{4}{3}L(\frac{h}{2}) - \frac{1}{3}L(h) - k_4\frac{1}{4}h^4 + O(h^6). \tag{6}$$

$$L = L_1(\frac{h}{2}) = \frac{4}{3}L(\frac{h}{4}) - \frac{1}{3}L(\frac{h}{2}) - k_4\frac{1}{4}(\frac{h}{2})^4 + O(h^6) \tag{7}$$

Multiply the second equation above by 16 and subtract the first equation from it, finally divide both sides by 15 to obtain:

$$L = L_2(h) = \frac{16}{15}L_1(\frac{h}{2}) - \frac{1}{15}L_1(h) + O(h^6) \tag{8}$$

Therefore we get:

$$L_2(h) = \frac{16}{15}L_1(\frac{h}{2}) - \frac{1}{15}L_1(h) = L_1(\frac{h}{2}) + \frac{1}{15}\cdot(L_1(\frac{h}{2}) - L_1(h)) \tag{9}$$

$\rightarrow a = \frac{1}{15}$

Question 5: Romberg Integration

a) ☐ $\frac{5}{3}$
☐ $\frac{53}{43}$
■ $\frac{51}{45}$
☐ $\frac{7}{6}$

With provided values: $I_0^1 = 1$, $I_0^2 = \frac{3}{2}$, $I_0^4 = \frac{5}{4}$ and with the use of formula from lecture:

$$I_k^n = \frac{4^k I_{k-1}^{2n} - I_{k-1}^n}{4^k - 1} \tag{10}$$

we get $I_1^1 = \frac{5}{3}$ and $I_1^2 = \frac{7}{6}$. Therefore we can also calculate I_2^1, which is equal to $\frac{51}{45}$.This is the best estimate of the integral, which one can find in this case.

b) ■ Trapezoidal Rule
☐ Midpoint Rule
☐ Simpson's Rule
☐ any Newton-Cotes formula

Romberg integration combines the Richardson Extrapolation with Trapezoidal Rule.

c) ☐ True
■ False

Only order of accuracy can be an even number if one uses Romberg integration with trapezoidal rule.

Question 6: Gaussian Quadrature

a) In the method of undetermined coefficients we approximate $\int_a^b f(x)dx$ as $c_1 f(a) + c_2 f(b)$. This approximation is exact for functions f of degree:

- ■ 1
- ☐ 2
- ☐ 3
- ☐ 4

b) Gaussian Quadrature is used to find the optimal number of quadrature points given a fixed interval $[a, b]$.

- ☐ True
- ■ False

c) Given an interval $[a, b] \; \forall \; a, b$ with $a < b$, the 3^{rd} order Gaussian quadrature integration points are $z = [0.7745966692, 0.0, -0.7745966692]$ for the approximation of $I = \int_a^b f(x)dx$.

- ☐ True
- ■ False

d) In Hermite interpolation, given data points x_i, y_i and y_i', the goal is to find a polynomial $f(x)$ of degree $2n-1$ that satisfies $y_i = f(x_i)$ and $y_i' = f'(x_i)$. For this $f(x)$ is expressed as:

$$f(x) = \sum_{k=1}^{n} U_k(x) y_k + \sum_{k=1}^{n} V_k(x) y_k^{'}$$

In order to satisfy the constraints on the data points and their derivatives, the polynomials $U_k(x)$ and $V_k(x)$ must have the following properties:

- ☐ $U_k(x_j) = 0$, $U_k^{'}(x_j) = \delta_j k$, $V_k(x_j) = 0$, $V_k^{'}(x_j) = \delta_j k$
- ■ $U_k(x_j) = \delta_j k$, $U_k^{'}(x_j) = 0$, $V_k(x_j) = 0$, $V_k^{'}(x_j) = \delta_j k$
- ☐ $U_k(x_j) = 0$, $U_k^{'}(x_j) = \delta_j k$, $V_k(x_j) = \delta_j k$, $V_k^{'}(x_j) = 0$
- ☐ $U_k(x_j) = \delta_j k$, $U_k^{'}(x_j) = 0$, $V_k(x_j) = \delta_j k$, $V_k^{'}(x_j) = 0$

e) To find the optimal quadrature points for $I = \int_a^b f(x)dx \approx c_1 f(x_1) + c_2 f(x_2)$, the approximation must be exact for the integration of any polynomial of order p. Select the highest possible value for p:

- ☐ 1
- ☐ 2
- ■ 3
- ☐ 4

Question 7: Bayesian Inference

a) We denote M by our model D as our data and Θ as the parameters of our model. The posterior distribution for the parameters is given by

$$P(\Theta|D, M) \tag{11}$$

Using Bayes' rule to estimate the value of the posterior, which are the quantities we need to specify?

- ☐ The prior $P(D|M)$, the likelihood $P(D|\Theta, M)$ and the evidence $P(\Theta|M)$
- ☐ The prior $P(\Theta|D)$, the likelihood $P(M|\Theta, D)$ and the evidence $P(M|D)$
- ■ The prior $P(\Theta|M)$, the likelihood $P(D|\Theta, M)$ and the evidence $P(D|M)$

b) The Laplace Approximation of an arbitrary probability distribution $P(\Theta)$ is based on a Taylor expansion of its logarithm

$$L(\Theta) = \log(P(\Theta)) \approx L(\Theta') + \frac{\partial L}{\partial \Theta}\bigg|_{\Theta'} (\Theta - \Theta') + \frac{1}{2}\frac{\partial^2 L}{\partial \Theta^2}\bigg|_{\Theta'} (\Theta - \Theta')^2 + \mathcal{O}\left[(\Theta - \Theta')^3\right] \quad (12)$$

In order to continue we

- ☐ ... exponentiate the result giving us a Gaussian approximation with variance

$$\sigma^2 = -\left(\frac{\partial^2 L}{\partial \Theta^2}\bigg|_{\Theta^*}\right)^{-1}$$

and mean located at the optimum Θ^*.

- ■ ... evaluate the Taylor expansion at the optimum $\Theta' = \Theta^*$ and exponentiate the result giving us a Gaussian approximation with variance

$$\sigma^2 = -\left(\frac{\partial^2 L}{\partial \Theta^2}\bigg|_{\Theta^*}\right)^{-1}$$

and mean located at the optimum Θ^*

- ☐ ... evaluate the Taylor expansion at the optimum $\Theta' = \Theta^*$ and exponentiate the result giving us a Gaussian approximation with variance

$$\sigma^2 = \left(\frac{\partial^2 L}{\partial \Theta^2}\bigg|_{\Theta^*}\right)^{-1}$$

and mean located at the optimum Θ^*

Numerical Problems

Question 8: One dimensional nonlinear solver [3 Points]

For a one-dimensional nonlinear function $f(x)$, we may use iterative numerical schemes to find its root, that is, find x^* so that $f(x^*) = 0$. Please apply Newton method to compute $\sqrt[3]{7}$. You may stop the iteration when the step size $\text{tol} = \|x^{k-1} - x^k\|$ is lower than 10^{-2}. To perform the calculation you might find the following table useful:

fraction	1/12	23/12	$(23/12)^2$	$3 \times (23/12)^2$	$(23/12)^3$	41/11021
float	0.083	1.917	3.675	11.021	7.041	0.004

Solution: It is equivalent to find zero of $f(x) = x^3 - 7$, where $f'(x) = 3x^2$. (1 point)

We use an initial integer guess $x_0 = 2$, as it gives $f(x_0) = x_0^3 - 7 = 1$, being close to zero. Therefore,

1. $x_0 = 2, f(x_0) = 1, f'(x_0) = 12$. (1 point)
2. $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 2 - 1/12 = 23/12 \approx 1.917$, $f(x_1) = (23/12)^3 - 7 \approx 0.041$, $f'(x_1) = 3 \times (23/12)^2 \approx 11.021$. (0.5 point)
3. $x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \approx 1.917 - 0.041/11.021 \approx 1.913$. (0.5 point)

We observe that x_2 does not change the second decimal digit of x_1. From calculator the true value of $\sqrt[3]{7} \approx 1.913$.

Question 9: Curvature of a carrot [12 Points]

You are trying to distinguish carrots from apples. For this, you take a photo of the object and detect the contours of it. Then you compute the curvature along the contours, if the curvature is higher than a certain threshold you decide that the object is a carrot. Else you classify it as an apple (see figure 2 for an illustration).

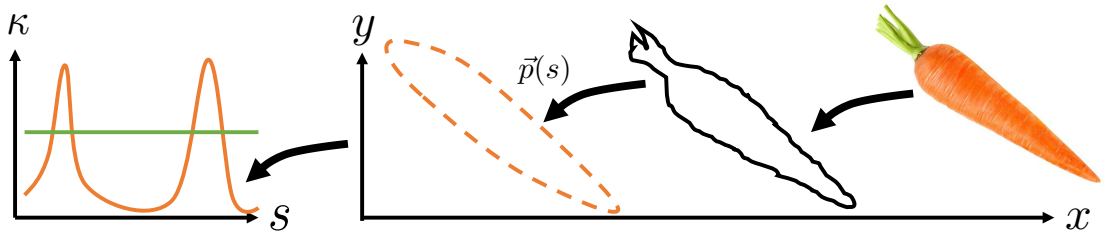


Figure 2: First the noisy contour of the image of the carrot is extracted. Then a parametrization of the contour is performed. Finally compute the curvature κ with respect to the parametrization variable s. The green line in the curvature plot approximatly depicts the curvature of an apple.

Since the contours can be very noisy you decide to fit cubic splines. However, first you need to parametrize it by introducing s, the distance along the contour:

$$\vec{p}(s) = (x(s), y(s))$$

The starting point is irrelevant since you are only interested in the peak curvature. Note that you have to fit cubic splines to both coordinates, independently.

a) Come up with boundary conditions for the cubic splines of each coordinate considering the fact that the curvature

$$\kappa(s) = \frac{|x'(s)y''(s) - y'(s)x''(s)|}{(x'^2(s) + y'^2(s))^{3/2}} \tag{13}$$

has to be continuous at s_1 and s_N, since s loops around the object.

Solution: In order for κ to be continuous, the following constraints need to be enforced:

$S'^x_1(s_1) = S'^x_{N-1}(s_N)$

$$S_1''^x(s_1) = S_{N-1}''^x(s_N)$$
$$S_1'^y(s_1) = S_{N-1}'^y(s_N)$$
$$S_1''^y(s_1) = S_{N-1}''^y(s_N)$$

Note that the the notations has not been introduced yet. As long as it is clear that the first and second derivatives at the end points have to be equal to the second derivatives at the starting point the points is assigned. (2 Point)

Let us assume you recover 4 points from your objects contour, $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)\}$. Please note that $(x_1, y_1) = (x_4, y_4)$, due to the loop. The corresponding s_i are are s_1, s_2, s_3, s_4. Each cubic spline writes as

$$S_i^j(s) = a_{i0}^j + a_{i1}^j s + a_{i2}^j s^2 + a_{i3}^j s^3$$

where j corresponds to the coordinate, either x or y, and $i = 1, 2, 3$ to the respective interval.

b) How many unknowns do we have to find in order to recover the full parametrization?

Solution: The cubic splines are required per coordinate, therefore a total of $2 \cdot 3 \cdot 4 = 24$. (1 Point).

c) Write down all constraints for the x-coordinate. The number of constraints should match the number of unknown parameters for the x-coordinate (half of the total number of unknowns). Use the generic formulation of the spline, $S_i^x(s)$, $S_i'^x(s)$, and $S_i''^x(s)$.

Solution:

$$S_1(s_1) = x_1$$
$$S_1(s_2) = x_2$$
$$S_2(s_2) = x_2$$
$$S_2(s_3) = x_3$$
$$S_3(s_3) = x_3$$
$$S_3(s_4) = x_4$$
$$S_1'(s_2) = S_2'(s_2)$$
$$S_2'(s_3) = S_3'(s_3)$$
$$S_1''(s_2) = S_2''(s_2)$$
$$S_2''(s_3) = S_3''(s_3)$$
$$S_1'(s_1) = S_3'(s_4)$$
$$S_1''(s_1) = S_3''(s_4)$$

Please note that x has been dropped from the notation. Also consider the fact that $(x_1, y_1) = (x_4, y_4)$, replacing them is not wrong. (3 Point)

d) Now expand the functions $S_i^x(s), S_i'^x(s), S_i''^x(s)$ and write out all the constraints for the x-coordinates.Bring them to a format $A_x \vec{a}_x = \vec{b}_x$. Where $\vec{a}_x$ contains all the unknowns a_{i0}^x, a_{i1}^x, a_{i2}^x, a_{i3}^x for $i = 1, 2, 3$ and A_x is a matrix and $\vec{b}_x$ is a vector that you are asked to construct.

Solution: All constraints written out are:

$$
\begin{aligned}
a_{10} + a_{11}s_1 + a_{12}s_1^2 + a_{13}s_1^3 &= x_1 \\
a_{10} + a_{11}s_2 + a_{12}s_2^2 + a_{13}s_2^3 &= x_2 \\
a_{20} + a_{21}s_2 + a_{22}s_2^2 + a_{23}s_2^3 &= x_2 \\
a_{20} + a_{21}s_3 + a_{22}s_3^2 + a_{23}s_3^3 &= x_3 \\
a_{30} + a_{31}s_3 + a_{32}s_3^2 + a_{33}s_3^3 &= x_3 \\
a_{30} + a_{31}s_4 + a_{32}s_4^2 + a_{33}s_4^3 &= x_4 \\
a_{11} + 2a_{12}s_2 + 3a_{13}s_2^2 - a_{21} - 2a_{22}s_2^2 - 3a_{23}s_2^2 &= 0 \\
a_{21} + 2a_{22}s_3 + 3a_{23}s_3^2 - a_{31} - 2a_{32}s_3^2 - 3a_{33}s_3^2 &= 0 \\
2a_{12} + 6a_{13}s_2 - 2a_{22} - 6a_{23}s_2 &= 0 \\
2a_{22} + 6a_{23}s_3 - 2a_{32} - 6a_{33}s_3 &= 0 \\
a_{11} + 2a_{12}s_1 + 3a_{13}s_1^2 - a_{31} - 2a_{32}s_4^2 - 3a_{33}s_4^2 &= 0 \\
2a_{12} + 6a_{13}s_1 - 2a_{32} - 6a_{33}s_4 &= 0
\end{aligned}
$$

(2 Point) This can be written in the mentioned matrix format with

$$
A_x = \begin{pmatrix}
1 & s_1 & s_1^2 & s_1^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & s_2 & s_2^2 & s_2^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & s_2 & s_2^2 & s_2^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & s_3 & s_3^2 & s_3^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & s_3 & s_3^2 & s_3^3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & s_4 & s_4^2 & s_4^3 \\
0 & 1 & 2s_2 & 3s_2^2 & 0 & -1 & -2s_2^2 & -3s_2^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 2s_3 & 3s_3^2 & 0 & -1 & -2s_3^2 & -3s_3^2 \\
0 & 0 & 2 & 6s_2 & 0 & 0 & -2 & -6s_2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 6s_3 & 0 & 0 & -2 & -6s_3 \\
0 & 1 & 2s_1 & 3s_1^2 & 0 & 0 & 0 & 0 & 0 & -1 & -2s_4 & -3s_4^2 \\
0 & 0 & 2 & 6s_1 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -6s_4
\end{pmatrix}
$$

and

$$
\vec{a}_x = \begin{pmatrix}
a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \\ a_{20} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{20} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{30} \\ a_{31} \\ a_{32} \\ a_{33}
\end{pmatrix}, \quad \vec{b}_x = \begin{pmatrix}
x_1 \\ x_2 \\ x_2 \\ x_3 \\ x_3 \\ x_4 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0
\end{pmatrix}
$$

Please note that the order of the rows can change, but the $\vec{a}_x$ has to be clearly defined. If this is not the case assume that they a's are in order. (2 Point)

e) At this point you have parametrized the contour of your object. You would like to know curvature $\kappa(s)$ in order to be able to classify the object later on. Assume that you found $a_{10}^x = 2$, $a_{11}^x = 0$, $a_{12}^x = -1$, $a_{13}^x = 0$, $a_{10}^y = 0$, $a_{11}^y = 2$, $a_{12}^y = 0$, and $a_{13}^y = -1/3$. Compute the curvature based on equation 13 of the arch between s_1 and s_2, where $s \ll 1$ (make appropriate simplifications clearly visible).

Solution: There are two approaches valid:

1. The resulting splines are Taylor approximations of $2\cos$ and $2\sin$, therefore the curvature is $1/R = 1/2$. (2 point)
2. Compute the required terms as:

   $S_1^x(s) = 2 - s^2$
   $S_1^y(s) = 2s - 1/3s^3$
   $S_1'^x(s) = -2s$
   $S_1'^y(s) = 2 - s^2$
   $S_1''^x(s) = -2$
   $S_1''^y(s) = -2s$

   (1 point) Then plug them into the curvature term and get rid of all terms higher than second order. The result is $\kappa = 1/2$. (1 point)

Question 10: Orthonormal functions [5 Points]

Data is generated according to the following model:

$$y = 2x + 5x^2$$

where x is drawn from a uniform distribution on the interval $[0, 1]$, $x \sim \mathcal{U}([0, 1])$. You are going to approximate the model with a set of orthonormal functions. Since you are dealing with a probabilistic model, an appropriate inner product has to be chosen.

$$\langle h, q \rangle = \mathbb{E}_{p(x)}[hq] = \int h(x)q(x)p(x)\mathrm{d}x$$

a) Compute a set of orthonormalized function from $\tilde{\phi}_1 = 1$ and $\tilde{\phi}_2 = x$ using Gram-Schmidt.

Solution: The orthonormal function of $\tilde{\phi}_1$ is $\phi_1 = 1$, since it already has unit length wrt. the inner product. The orthogonal function of $\tilde{\phi}_2$ is $x - 1/2$, given by Gram-Schmidt. To obtain the normalization constant compute $\sqrt{\mathbb{E}[(x-1/2)^2]} = \frac{1}{2\sqrt{3}}$. The resulting orthonormal function is $\phi_2 = 2\sqrt{3}(x - 1/2)$. (1.5 Points)

b) You now want to approximate y with the two orthonormal functions. Compute α_1 and α_2 and write down the resulting $y \approx \alpha_1\phi_1(x) + \alpha_2\phi_2(x)$.

Solution: Compute the inner products $\langle y, \phi_1 \rangle$: $\int_0^1 y(x)\phi_1(x)\mathrm{d}x = 8/3$ and $\int_0^1 y(x)\phi_2(x)\mathrm{d}x = \frac{7}{2\sqrt{3}}$, giving $y(x) \approx 8/3 + 7(x - 1/2)$ (1.5 Points)

c) Assume that you have an additional orthonormalized function ϕ_3. Show the previously computed α's do not change for the new approximation

$$y \simeq \sum_{i=1}^{3} \alpha_i \phi_i(x) \tag{14}$$

Hint: Take advantage of the fact that the inner product is linear, i.e. $\langle c \cdot h, q \rangle = c \cdot \langle h, q \rangle$, where c is a constant, and $\langle h + f, q \rangle = \langle h, q \rangle + \langle f, q \rangle$.

Solution: (2 Points) We want to show that the coefficients do not depend on any other basis-function. Therefore we need linearity of the scalar product and the orthonormality of the basis functions:

$$\begin{aligned}
\langle y, \phi_i \rangle &= \langle \alpha_1\phi_1 + \alpha_2\phi_2 + \alpha_3\phi_3, \phi_i \rangle \\
&= \alpha_1\langle \phi_1, \phi_i \rangle + \alpha_2\langle \phi_2, \phi_i \rangle + \alpha_3\langle \phi_3, \phi_i \rangle \\
&= \alpha_1\delta_{i1} + \alpha_2 \cdot \delta_{i2} + \alpha_3 \cdot \delta_{i3} \\
&= \alpha_i
\end{aligned}$$

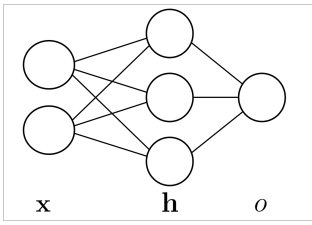
Question 11: Backpropagation [12 Points]

A network is built with two input neurons, one hidden layer of 3 neurons and an output layer with a single neuron. Both layers have a rectified linear unit (RELU) activation functions. This network does not have any biases.
The rectified linear unit (RELU) is defined as

$$\phi(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

a) Sketch the graph of the network, make sure to label the input $\mathbf{x}$, the hidden layer $\mathbf{h}$ and the output o.

Solution:
(1 Point)



b) Write the relations between the components of the input $\mathbf{x}$, the hidden layer $\mathbf{h}$ and the output o.

Solution: The relation between the input and the hidden layer is given as :

$$h_i = \phi(z_i^H) \text{ with } z_i^H = \sum_{j=1}^{2} W_{i,j}^H x_j \ \forall i = 1, 2, 3 \qquad \text{(1 Point)}$$

The relation between the hidden layer and the output is given as:

$$o = \phi(z^O) \text{ with } z^O = \sum_{j=1}^{3} W_j^O h_j \qquad \text{(1 Point)}$$

Variations on the notations between using different subscripts in place of H and O are accepted. Variations on the i and j index accepted as long as it makes sense.

c) Compute the forward pass (i.e the output o) for an input sample $\mathbf{x}^1 = (1, 2)$. The hidden weight matrix $\mathbf{W}^H$ and the output weight matrix $\mathbf{W}^O$ are initialized as follows:

$$\mathbf{W}^H = \begin{bmatrix} -3 & 1 \\ 3 & 1 \\ 2 & 2 \end{bmatrix}, \ \mathbf{W}^O = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \qquad (16)$$

Solution: Following the previously derived relations the hidden layer is computed as :

$$\mathbf{h} = \phi(\mathbf{z}^H) = \begin{bmatrix} 0 \\ 5 \\ 6 \end{bmatrix} \text{ with } \mathbf{z}^H = \begin{bmatrix} -3 & 1 \\ 3 & 1 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} -1 \\ 5 \\ 6 \end{bmatrix} \qquad \text{(1 Point)}$$

The output is computed as:

$$o = \phi(z^O) = 16 \text{ with } z^O = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 5 \\ 6 \end{bmatrix} = 16 \qquad \text{(1 Point)}$$

d) Compute the error $E = \frac{1}{2}(\mathbf{t} - \mathbf{o})^2$ given the output o computed in the previous subquestion and the target $\mathbf{t}^1 = 14$.

Solution: The error with respect to the L2 norm and given the target $\mathbf{t_1} = 14$ and the previously computed output $o = 16$ is given as :

$$E = \frac{1}{2}(o - t)^2 = \frac{1}{2}(16 - 14)^2 = 2 \qquad \text{(0.5 Point)}$$

e) Compute the gradient $\frac{\partial E}{\partial W_2^O}$ of the error with respect to weight W_2^O, the second weight of the output layer (use the chain rule).

Solution: The gradient of the error with respect to the second component of the output weight matrix W_2^O is given by the chain rule as:

$$\frac{\partial E}{\partial W_2^O} = \frac{\partial E}{\partial o} \frac{\partial o}{\partial z^O} \frac{\partial z^O}{\partial W_2^O} \qquad \text{(1 Point)}$$

Starting from the general formulation of the error, the gradient with respect to the output is

$$\frac{\partial E}{\partial o} = \frac{\partial}{\partial o} \frac{1}{2}(t - o)^2 = \frac{1}{2} 2(t - o) = (o - t) \qquad \text{(1 Point)}$$

Replacing with the values of the output and target we get:

$$E = (o - t) = (16 - 14) = 2 \qquad \text{(0.5 Point)}$$

The gradient of the output with respect to z^O is the derivative of the RELU activation function:

$$\frac{\partial o}{\partial z^O} = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases}. \qquad \text{(1 Point)}$$

The gradient of z^O with respect to the weight W_2^O is given as

$$\frac{\partial z^O}{\partial W_2^O} = \frac{\partial}{\partial W_2^O} \sum_{i=1}^{3} W_i^O h_i = h_2 = 5. \qquad \text{(1 Point)}$$

Replacing in the first expression we get:

$$\frac{\partial E}{\partial W_2^O} = \frac{\partial E}{\partial o} \frac{\partial o}{\partial z^O} \frac{\partial z^O}{\partial W_2^O} = 2 * 1 * 5 = 10. \qquad \text{(1Point)}$$

f) Compute the update weight $W_{2,new}^O$ based on the old weight W_2^O using one step of gradient descent with the given learning rate $\eta = 0.1$ and the gradient $\frac{\partial E}{\partial w_2^0}$ computed in the previous subquestion.

Solution: The update for any weight $W_{i,j}^k$ is given as:

$$W_{i,j}^{k,new} = W_{i,j}^k + \delta W_{i,j}^k \text{ with, } \delta W_{i,j}^k = -\eta \frac{\partial E}{\partial W_{i,j}^k} \qquad (17)$$

In this case we get:

$$W_2^{O,new} = W_2^O - \eta \frac{\partial E}{\partial W_2^O} = 2 - 0.1 * 10 = 1. \qquad \text{(1 Point)}$$

Question 12: Open Newton-Cotes formula [6 Points]

The Newton-Cotes formula allows us to derive Quadrature rules by approximating the objective function by Lagrange polynomials. There exists two types of Newton-Cotes formulas, the "closed" type in which the end points of the interval are included in the quadrature (which you have learned in class), and the "open" type in which the end points are not included.
For a given order M the quadrature points for "closed" type are expressed as

$$x_i = a + i\frac{b-a}{M}, \qquad i = 0, ..., M$$

For order M the "closed" type are expressed as

$$x_i = a + i\frac{b-a}{M}, \qquad i = 1, ..., M-1$$

a) Derive the trapezoidal method of "open" type with $M = 3$ to approximate an arbitrary integral

$$I = \int_a^b f(x)\mathrm{d}x$$

Solution: For $M = 3$, $[a, b]$ and for "open" type formula one obtain (1 point):

$$\Delta_i = \frac{b-a}{3} \quad , \quad x_1 = a + \Delta_i = \frac{2a+b}{3} \quad , \quad x_2 = a + 2 \cdot \Delta_i = \frac{2b+a}{3} \tag{18}$$

and the Lagrange interpolants through these points are (1 point):

$$l_1^2(x) = \frac{x - x_2}{x_1 - x_2} \tag{19}$$

$$l_2^2(x) = \frac{x - x_1}{x_2 - x_1} \tag{20}$$

With this we can compute coefficients C_k^M, i.e. C_1^2 and C_2^2 (2 point):

$$C_1^2 = \frac{1}{\Delta_i} \int_a^b l_1^2(x)\,\mathrm{d}x = \frac{3}{2} \tag{21}$$

$$C_2^2 = \frac{1}{\Delta_i} \int_a^b l_2^2(x)\,\mathrm{d}x = \frac{3}{2} \tag{22}$$

Therefore the approximation is given as:

$$I \approx \frac{b-a}{3} \cdot (\frac{3}{2} f(\frac{2a+b}{3}) + \frac{3}{2} f(\frac{2b+a}{3})) \tag{23}$$

b) Calculate the integral

$$J = \int_0^{\pi/2} \sin(x)dx$$

using the trapezoidal method of "open" type derived in the previous subquestion. *Hint:* You may find the following values useful:

x	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
$\sin(x)$	0	$1/2$	$1/\sqrt{2}$	$\sqrt{3}/2$	1	0	-1	0

Solution: (1 point)

$$J \approx \frac{\frac{\pi}{2} - 0}{3} \cdot (\frac{3}{2}\sin(\frac{\pi}{6}) + \frac{3}{2}\sin(\frac{\pi}{3})) = \frac{3 + 3\sqrt{3}}{24}\pi \tag{24}$$

c) Compare the result to the trapezoidal method of "closed "type, which is given by

$$J \approx \frac{\Delta x}{2}[f(a) + f(b)]$$

Compare the result to the previous subquestion and discuss the difference with respect to the exact solution $J = 1$.
Hint: You may find the following values useful:

x	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$	$\frac{\pi}{2}$	π	$\frac{3\pi}{2}$	2π
$\sin(x)$	0	$1/2$	$1/\sqrt{2}$	$\sqrt{3}/2$	1	0	-1	0

Further you may use $\pi \approx 3$, $\sqrt{2} \approx 1.5$ and $\sqrt{3} \approx 1.75$.

Solution: (1 point) Here, for the trapezoidal method of closed type we have $a = 0$, $b = \frac{\pi}{2}$ and $\Delta x = b - a = \frac{\pi}{2}$

$$J \approx \frac{\Delta x}{2}[\sin(a) + \sin(b)] = \frac{\pi}{4} \tag{25}$$

Now comparing to exact solution :

$$\varepsilon_{open} = |\frac{3 + 3\sqrt{3}}{24}\pi - 1| = 0.073 \tag{26}$$

$$\varepsilon_{closed} = |\frac{\pi}{4} - 1| = 0.215 \tag{27}$$

with assumption $\sqrt{3} \approx 1.7$ and $\pi \approx 3$:

$$\varepsilon_{open} = |\frac{3 + 3 \cdot 1.7}{24} \cdot 3 - 1| = \frac{1}{80} \tag{28}$$

$$\varepsilon_{closed} = |\frac{3}{4} - 1| = \frac{1}{4} \tag{29}$$

Question 13: Monte Carlo Sampling [12 Points]

In statistical mechanics we aim to derive macroscopic thermal quantities based on probabilistic properties of the underlying microscopic systems. In the following we will regard the canonical ensemble, which is a mechanical system with fixed volume V and particle number N, which is in thermal equilibrium with a heat bath of temperature T. In this case it can be shown that the probability to find a particle with momentum $\vec{p} \in \mathbb{R}^3$ and position $\vec{q} \in \mathbb{R}^3$ corresponding to an energy $H(\vec{p}, \vec{q})$ is given by the Boltzmann (or Gibbs) distribution

$$P(\vec{p}, \vec{q}) = \frac{1}{Z} \exp\left(-\frac{H(\vec{p}, \vec{q})}{kT}\right), \tag{30}$$

where k is the Boltzmann constant and the normalization factor is the partition function

$$Z = C \int \exp\left(-\frac{H(\vec{p}, \vec{q})}{kT}\right) \mathrm{d}\vec{p}_1 \cdots \mathrm{d}\vec{p}_N \mathrm{d}\vec{q}_1 \cdots \mathrm{d}\vec{q}_N. \tag{31}$$

Here C is some constant taking into account the phase space volume and permutations of the indistinguishable particles we consider. If this integral can be solved analytically we can derive the free energy of the system as

$$F = -kT \ln(Z) \tag{32}$$

From this the macroscopic quantities can be derived. In most cases the integration can not be performed analytically and therefore we have to use numerical methods to approximate it.

a) Please write down the formula allowing you to integrate a function over an $6N$ dimensional space using numerical quadrature. Is this feasible if you assume that $N = 10^{23}$? Explain the associated problem using the fact that the Trapezoidal rule is second order accurate by deriving the error with respect to the number of sampling points.

Solution: We can solve multi-dimensional integrals using a cartesian product of one-dimensional quadrature rules

$$I \approx \sum_{\substack{i_1=1 \\ \cdots \\ i_d=1}}^{n} \tilde{w}_{i_1,\ldots,i_d} f\left(x_{i_1}^{(1)}, \ldots, x_{i_d}^{(d)}\right), \quad \text{with} \quad \tilde{w}_{i_1,\ldots,i_d} = \prod_{r=1}^{d} w_{i_r} \qquad \text{(1 Point)}$$

We know that the Trapezoidal rule in one dimension is second order accruate, i.e.

$$I - I_T = \mathcal{O}[h^2] = \mathcal{O}\left[\left(\frac{L}{n}\right)^2\right] = \mathcal{O}\left[\left(\frac{1}{n^2}\right)\right] \qquad \text{(1 Point)}$$

In the case we are integrating over a d dimensional space the total number of points we need to use is $N = n^d$, thus we can rewrite the order for the error as

$$I - I_T = \mathcal{O}\left[\left(\frac{1}{N^{2/d}}\right)\right] \qquad \text{(1 Point)}$$

We realize that the error goes exponentially like the dimension. In the case of a statistical ensemble with $d = 6 \cdot 10^{23}$ this is unfeasable.

b) Write down the formula to compute a $6N$ dimensional integral with Monte Carlo Integration. As this is a stochastic method the estimate we obtain is a random variable. Thus we have to estimate the error ϵ. A common way to do that goes via the standard deviation

$$\epsilon = \sqrt{\mathrm{Var}(\langle f \rangle_M)} \qquad (33)$$

Show that the resulting error scales as $1/\sqrt{M}$ for Monte Carlo integration.
Hint: Use the fact that your samples are independent (i.e. $\mathbb{E}[f(x_i)f(x_j)] = \mathbb{E}[f(x_i)]\mathbb{E}[f(x_j)]$)

Solution: Monte Carlo integration approximates the integral as

$$\langle f \rangle_M = \frac{|\Omega|}{M} \sum_{i=1}^{M} f\left(\vec{x}_i\right) \qquad \text{(1 Point)}$$

Given the error for Monte Carlo Integration we find

$$\begin{aligned}
\varepsilon_M^2 = \mathrm{Var}\left[\langle f\rangle_M\right] &= \left\langle \langle f\rangle_M^2 \right\rangle - \left\langle \langle f\rangle_M \right\rangle^2 && \text{(1 Point)}\\
&= \frac{1}{M^2}\sum_{i,j=1}^{M}\left(\mathbb{E}\left[f\left(\mathbf{x}_i\right) f\left(\mathbf{x}_j\right)\right] - \langle f\rangle^2\right) && \text{(1 Point)}\\
&= \frac{1}{M^2}\sum_{i=1}^{M}\left(\mathbb{E}\left[f\left(\mathbf{x}_i\right)^2\right] - \langle f\rangle^2\right)\\
&+ \frac{1}{M^2}\sum_{\substack{i,j=1\\ i\neq j}}^{M}\left(\underbrace{\mathbb{E}\left[f\left(\mathbf{x}_i\right) f\left(\mathbf{x}_j\right)\right]}_{=[\mathbb{E}f(\mathbf{x}_i)]\mathbb{E}[f(\mathbf{x}_j]]=\langle f\rangle^2} - \langle f\rangle^2\right) && \text{(1 Point)}\\
&= \frac{1}{M^2}\sum_{i=1}^{M}\left(\left\langle f^2\right\rangle - \langle f\rangle^2\right) && \text{(1 Point)}\\
&= \frac{1}{M^2} M\left(\left\langle f^2\right\rangle - \langle f\rangle^2\right) = \frac{\langle f^2\rangle - \langle f\rangle^2}{M} = \frac{\mathrm{Var}[f]}{M} && \text{(1 Point)}
\end{aligned}$$

Thus we realize that the error does not scale with the dimension of the problem, but is always

$$I - I_M = \mathcal{O}\left(\frac{1}{\sqrt{M}}\right) \qquad \text{(1 Point)}$$

Thus starting from dimension 4 Monte-Carlo integration is more accurate then the Trapezoidal rule.

Monte Carlo methods depend on our ability to sample from the probability distribution governing our problem. One way to generate samples from an arbitrary distribution goes via inverse transform sampling. This method is based on the fact that samples of a random variable x with cumulative distribution function $F_X(x)$ can be obtained via samples from an uniform distribution $u \sim \mathcal{U}([0,1])$ via the following relation

$$x = F_X^{-1}(u) \tag{34}$$

Here F_X^{-1} denotes the inverse of the cumulative distribution function.

c) Given the exponential distribution

$$p(q) = \lambda e^{-\lambda x}, \tag{35}$$

derive the expression allowing you to generate samples using the inverse transform method.

Solution: Given the pdf for the exponential distribution it is straight forward to compute the cdf as

$$F(x) = \int_0^x p(x)dx = \int_0^x \lambda e^{-\lambda x}dx = 1 - e^{-\lambda x} \qquad \text{(1 Point)}$$

setting $F_X(x) = u$ we realize that this equation can readily be inverted and allows us to generate samples from the exponential distribution $x^{(i)}$ starting from uniform random numbers $u^{(i)}$ via

$$x^{(i)} = -\frac{1}{\lambda}\ln\left(1 - u^{(i)}\right) \qquad \text{(1 Point)}$$

Pseudocode

Question 14: Linear Least Squares [2 Points]

In the year of 2050, one astronaut landed on a planet with environment quite similar to earth in the Alpha Centauri star system, which is the closest and 4.37 light-years away from our solar system. The first task performed by the astronaut is quite simple: the astronaut throws a ball into the sky and records accurately a sequence of height of the ball and and time elapse: (t_1, h_1), (t_2, h_2), (t_3, h_3), ... (t_N, h_N). Following Newton's law of motion, the astronaut tries to figure out the gravity/acceleration on this planet: $h = g/2t^2 + v_0 t + h_0$, where positive h is upwards, g is the gravity, v_0 and h_0 are initial velocity and position of the ball, respectively. The astronaut formulated the linear least squares for this problem and wrote a pseudocode as follows. To be more specific, the astronaut identified matrix A, unknown $\vec{x}$, and right hand side $\vec{h}$. Furthermore, the astronaut built the so-called normal equations and tried to solve them. Please help identify the mistakes/bugs made in the pseudocode.

Algorithm 1 Linear Least Squares: pseudocode

Input:

N, {number of records}
t, {vector containing time sequence}
h, {vector containing height sequence}

Output:

g, {gravity}
v_0, {initial velocity}
h_0, {initial height}

Steps:

$i \leftarrow 1$
while $i \leq N$ **do**
    $A[i, 1] \leftarrow t[i] * t[i]$ 1/2 missing (0.5 point)
    $A[i, 2] \leftarrow t[i]$
    $A[i, 3] \leftarrow 0$ should be 1 (0.5 point)
end while

Steps:

$A_T \leftarrow \text{transpose}(A)$
$B \leftarrow$ dot product(A, A_T) (A_T, A) (0.5 point)
$C \leftarrow$ dot product(A_T, h)

Steps:

$D \leftarrow$ matrix inverse(B)
$x \leftarrow$ dot product(C, D) (D, C) (0.5 point)
$g \leftarrow x[0]$
$v_0 \leftarrow x[1]$
$h_0 \leftarrow x[2]$

Question 15: Monte Carlo Integration [8 Points]

a) Write a pseudo code allowing you to integrate an arbitrary function using Monte Carlo Integration. Assume you are given a routine generating samples from an uniform distribution.

Solution:

Algorithm 2 Monte Carlo Integration

Input:(1 Point)

N, {number of samples to compute}
$\Omega = [x_-, x_+] \times [y_-, y_+]$, {Integration Domain}
$f(x)$, {Function to Integrate}

Output:(1 Point)

I, {Approximation of the integral}

Steps:

$K = 0$
for $i \leftarrow 1, \ldots, N$ **do** (1 Point)
  Sample $x_i \sim \mathcal{U}([x_-, x_+])$
  Sample $y_i \sim \mathcal{U}([y_-, y_+])$
  Compute $f(x_i)$.
  if $y_i < f(x_i)$ **then** (1 Point)
    $K = K + 1$
  end if
end for
$V = K/N|\Omega|$ (1 Point)
return V

b) You want to estimate how well your algorithm performs. Therefore you decide to regard the acceptance rate A (i.e. the number of accepted samples divided by the total number of samples). What changes do you have to do in your code to compute this measure? If you have enough space in your code from part a), please add these lines to your code. If not mark the associated places (using $\star$,# or similar) and write down the associated pseudo-codes on separate lines.

Solution (1 Point):

Algorithm 3 Monte Carlo Integration

Input:

N, {number of samples to compute}
$\Omega = [x_-, x_+] \times [y_-, y_+]$, {Integration Domain}
$f(x)$, {Function to Integrate}

Output:

I, {Approximation of the integral}

Steps:

$K = 0$
for $i \leftarrow 1, \ldots, N$ **do**
  Sample $x_i \sim \mathcal{U}([x_-, x_+])$
  Sample $y_i \sim \mathcal{U}([y_-, y_+])$
  Compute $f(x_i)$.
  if $y_i < f(x_i)$ **then**
    $K = K + 1$
  end if
end for
$A = K/N$
$V = A|\Omega|$
return V, A

c) Assuming you found that the acceptance rate is really low. What are the two possible variants to the naive Monte Carlo Integration introduced in the lecture? Which parts of the code have to be adjusted for these improvements?

Solution: We can implement rejection sampling where instead of sampling (x, y) uniformly we sample from a distribution X with pdf p_X, which multiplied by some constant λ covers the graph f (1 Point) or importance sampling, where we also replace the uniform sampling and replace $f \to f/p_x$ (1 Point).

Algorithm 4 Monte Carlo Integration

Input:

N, {number of samples to compute}
$\Omega = [x_-, x_+] \times [y_-, y_+]$, {Integration Domain}
$f(x)$, {Function to Integrate}

Output:

I, {Approximation of the integral}

Steps:

$K = 0$
for $i \leftarrow 1, \ldots, N$ **do**
  Sample $x_i \sim X$
  Sample $y_i \sim \mathcal{U}([y_-, y_+])$
  Compute $f(x_i)/p_X(x_i)$.
  if $y_i < f(x_i)/\lambda p_X(x_i)$ **then**
    $K = K + 1$
  end if
end for
$A = K/N$
$V = A|\Omega|$
return V, A

Good luck!