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

Spring semester 2018

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

Exam

Issued: August 8, 2018

Short Questions

Question 1: Linear Least Squares (4 points)

Let $\bar{x}$ be the linear least squares solution to the problem $A\vec{x} = \vec{b}$, and $\vec{E} = A\bar{x} - \vec{b}$ be the corresponding residual vector.

a) Describe the geometric interpretation of the least squares solution.

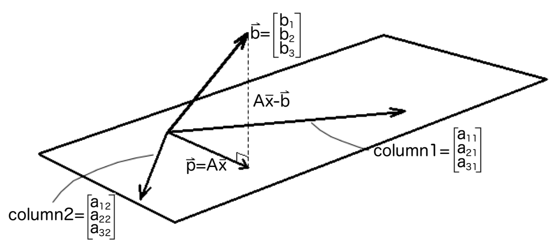

Figure 1: Least squares solution to a 3×2 least squares problem.

b) Let matrix A be equal to

$$A = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{pmatrix}.$$

Which of the following vectors is a possible value for $\vec{E}$, why?

☐ $\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$ ☑ $\begin{pmatrix} 0 \\ 1 \\ -2 \\ 1 \end{pmatrix}$ ☐ $\begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$ ☐ $\begin{pmatrix} -1 \\ -1 \\ -1 \\ -1 \end{pmatrix}$

Solution is (b). It is orthogonal to both columns of A.

Question 2: Lagrange interpolation (5 points)

State whether the following statements are true or false. Each correct answer is awarded one point, answers left blank are worth zero points. Each wrong answer is penalized with minus one point. The minimum score is zero.

a) For Lagrange interpolation of N data points, the degree of each basis function $L_i(x)$ is N.
False, it is of order N-1 (you leave out one point).

b) The Lagrange polynomial passes through every point of the given data set.
True.

c) The Lagrange polynomial is the polynomial of highest degree that assumes the value y_i at each point x_i of a data set $\{x_i, y_i\}, i = 1...N$.
False, it is the lowest order polynomial that passes through the set of N points (order N-1 in this case).

d) If $P(x)$ is a Lagrange polynomial interpolating the function $f(x) = \frac{1}{x}$ at the points $x_i = \{0, 1, 4\}$, then $P'(4) = \frac{-1}{16}$.
False, Lagrange interpolating polynomials do not guarantee that the derivatives will match.

e) If nodes x_k are equispaced over some interval $x \in [a, b]$, then the Lagrange interpolating function oscillates with greater amplitude near a and b than near the midpoint of the interval.
True. This is just Runges Phenomenon

Question 3: Neural networks (6 points)

Are the following statements true or false? Each correct answer is awarded one point, answers left blank are worth zero points. Each wrong answer is penalized with minus one point. The minimum score is zero.

a) Gradient descent based methods, such as back-propagation may get stuck in local minima.
The statement is true. Gradient descent based methods only move towards descent directions and may get stuck in local minima.

b) Increasing the number of hidden nodes in a feed-forward neural network with two layers and non-linear activations improves generalization.
The statement is false. Increasing the number of hidden nodes in a feed-forward neural network improves the representation ability of the network. It does not necessarily mean that generalization performance is also improved, since the network is more prone to overfitting.

c) Increasing the number of layers in a feed-forward neural network with non-linear activations makes it more prone to overfit.
The statement is true. Increasing the number of hidden layers in a feed-forward neural network improves the representation ability of the network. However, the network is more prone to overfitting as the number of weights rises and the network may memorize the training data-set.

d) Assume that you have two networks without activation functions (just linear hidden layers). The input and output layers of the neural networks are the same. The first one has three hidden layers, while the second only one. These two neural networks can approximate the same class of functions.

The statement is true. Assume that the input is x, both networks can approximate the class of linear functions of the input $f = Wx$ independent of the number of layers. For the one-dimensional input and output, having three layers with weights w_1, w_2, and w_3 is equivalent with the one-layered network with a weight equal to $w = w_1 \cdot w_2 \cdot w_3$. The number of layers does not matter.

e) During training of a neural network with gradient descent, shuffling the data of each batch before performing the training step on this batch can help the network avoid local minima and converge faster to an optimum.
The statement is false. Shuffling the order of the samples in a batch does not influence the optimization step. The gradient of the batch is averaged among the samples. As a consequence, the order does not matter.

f) Generalization is the property of a learning system to approximate the target output values for inputs that are not included in the testing data-set.
The statement is false. Generalization is the property of a learning system to approximate the target output values for inputs that are not included in the **training** data-set.

Question 4: Newton's method (3 points)

Show that the formula for determining the square root of a

$$x_n = \frac{1}{2}\left(x_{n-1} + \frac{a}{x_{n-1}}\right)$$

is a special case of Newton's iteration. Generalize the formula for the p-th root of a.

Applying the Newton's formula and using $f(x) = x^2 - a$, $f'(x) = 2x$ we obtain

$$x_n = x_{n-1} - \frac{x_{n-1}^2 - a}{2x_{n-1}} = \frac{1}{2}(x_{n-1} + \frac{a}{x_{n-1}})$$

For the p-th root of a we have $f(x) = x^p - a$, $f'(x) = px^{p-1}$ and thus

$$x_n = x_{n-1} - \frac{x_{n-1}^p - a}{px_{n-1}^{p-1}} = \frac{p-1}{p}(x_{n-1} + \frac{a}{(p-1)x_{n-1}^{p-1}})$$

Question 5: Numerical Integration (7 points)

You are asked to approximate the integral $\int_a^b f(x)$ by:

a) Deriving the one-point quadrature rule, starting with the expression $\int_a^b f(x)\,dx = c_1 f(x_1)$.
The one-point quadrature rule needs to give exact values for a linear polynomial of order one. As a result the expression $c_1 f(x_1)$ needs to be exact for both a constant and a linear in x form of the function $f(x)$. So, using as a constant $\int_{-1}^{1} 1dx$ and as a linear term $\int_{-1}^{1} xdx$, we obtain:

$$\begin{aligned} \int_a^b 1dx &= b - a = c_1 \\ \int_a^b xdx &= \frac{b^2 - a^2}{2} = c_1\, x_1 \end{aligned}$$

Upon which we obtain $x_1 = \dfrac{b+a}{2}$, which leads to the following expression:

$$\int_a^b f(x)\,dx = (b-a)\,f\left(\frac{b+a}{2}\right)$$

Given the formula obtained, you are requested to provide answers regarding:

b) A different name that is commonly given to the previously derived formula.

The formula is well-known as the midpoint or rectangular integration rule.

c) The appropriateness of the obtained formula for exact computation of $\int_0^{\pi/2} \cos x dx$.

Not appropriate. It cannot accurately integrate non-linear functions.

d) The expected maximum order of polynomial that could be accurately integrated if a Simpson's integration rule was used instead.

At most 3^{rd} order polynomials can be accurately integrated.

Question 6: Monte Carlo Integration (9 points)

Your task is to compute the electrostatic energy of an electric charge being moved in the electric field of a uniformly charged solid sphere:

$$E_{el} = -\int_{r_1}^{r_2} \frac{1}{r^2} dr,$$

where r is the radius from the center of the sphere, and r_1 and r_2 are constants and $r_2 > r_1$.

a) From a computational perspective does it make sense to use monte carlo integration? Briefly explain.

It doesn't make sense in a computational aspect, since we only have one variable.

b) We assume that we are drawing the radius from a random uniform distribution $r \sim uniform(r_1, r_2)$. Write down the expectation value of $\frac{-1}{r^2}$ over the distribution, you do not need to compute it.

$$\mathbb{E}\left[\frac{-1}{r^2}\right] = \frac{1}{r_2 - r_1} \int_{r_1}^{r_2} \frac{-1}{r^2} dr$$

c) Manipulate the expectation value as such that you reobtain the integral E_{el} you are trying to compute. Then write how you can approximate the integral if you only have a finite number of M samples.

$$I = (r_2 - r_1) \cdot \mathbb{E}\left[\frac{-1}{r^2}\right] \approx (r_2 - r_1) \cdot \frac{1}{M} \sum_{i=1}^{M} \frac{-1}{r_i^2}$$

You now try to compute the electrostatic energy through measurements. You have a device that measures the electric field at random distances. However, your measurements are not uniformly distributed but according to the following probability density function, due to the device:

$$P(r) \sim C \cdot r^2$$

Where C is a constant.

d) Calculate C such that equation is a valid probability density function. Explicitly show that the conditions are met.

Calculate $1 = C \int_{r_1}^{r_2} r^2 dr$ and solve for C. Then

$$C = \frac{3}{r_2^3 - r_1^3}$$

Since $r_2 > r_1$ we have that $p(r) \geq 0$

Numerical Problems

Question 7: Lagrange Interpolation (8 points)

You sample the function $f(x) = \frac{-2}{3}x^3 + \frac{5}{3}x^2 + 4x$ at the points $\{x_i\} = \{0, 1, 3\}$ and wish to compute the Lagrange Interpolating polynomial for this dataset.

a) Compute all the Lagrange basis functions for the above data.

The lagrange basis functions are given by:

$$\begin{aligned}
l_1(x) &= \frac{(x-1)(x-3)}{(0-1)(0-3)} \\
&= \frac{1}{3}(x^2 - 4x + 3) \\
l_2(x) &= \frac{(x-0)(x-3)}{(1-0)(1-3)} \\
&= \frac{-1}{2}(x^2 - 3x) \\
l_3(x) &= \frac{(x-0)(x-1)}{(3-0)(3-1)} \\
&= \frac{1}{6}(x^2 - x)
\end{aligned}$$

b) Construct the Lagrange interpolation of the above data.

The full lagrange interpolation function is achieved by taking linear combinations of the above computed basis functions weighted by their respective 'y' data points. First we compute the values of $f(x)$ at the points of interest:

$$
\begin{aligned}
f(0) &= 0 \\
f(1) &= \frac{-2}{3} + \frac{5}{3} + 4 \\
&= 5 \\
f(3) &= \frac{-2}{3}3^3 + \frac{5}{3}3^2 + 12 \\
&= -18 + 15 + 12 \\
&= 9
\end{aligned}
$$

The full Lagrange interpolation is then given by:

$$
\begin{aligned}
L(x) &= 0 \cdot l_1(x) + 5 \cdot l_2(x) + 9 \cdot l_3(x) \\
&= \frac{-5}{2}(x^2 - 3x) + \frac{3}{2}(x^2 - x) \\
&= 6x - x^2
\end{aligned}
$$

c) Interpolate the above data at $x = 2$ using your previous results and compute the interpolation error. How would your interpolation error change if you sampled a fourth point?

Evaluating the Lagrange interpolation at $x = 2$ yields a result of 8, while evaluating the true objective function at $x = 2$ yields a result of $\frac{28}{3}$. The resulting interpolation error is $\frac{4}{3}$. Naturally, we shouldn't expect this interpolation error to be zero since we are modeling a cubic function with a quadratic. Had we sampled a fourth point, then the resulting Lagrange interpolating function would have been cubic instead of quadratic, and we would have had a perfect fit of the objective function and zero interpolation error.

Question 8: Gram-Schmidt orthonormalization (5 points)

Assume you are given a set of functions $\varphi = \{x, \sin(x)\}$ on the interval $[-\pi, \pi]$.

a) Write down the appropriate condition for orthonormality between two functions ϕ_i and ϕ_j and orthonormalise the basis if necessary. *Hint: use integration by parts* $\int_a^b u(x)v'(x)dx = [u(x)v(x)]_a^b - \int_a^b u'(x)v(x)dx$ *and* $\int_{-\pi}^{\pi} \sin^2(x)dx = \pi$.

$$
\begin{aligned}
\langle \phi_i, \phi_j \rangle &= \delta_{i,j} \\
\langle x, \sin(x) \rangle &= \int_{-\pi}^{\pi} x \cdot sin(x) \, \mathrm{d}x \\
&= (-xcos(x))|_{-\pi}^{\pi} + \int_{-\pi}^{\pi} cos(x) \, \mathrm{d}x \\
&= -2\pi cos(\pi) + 0 \\
&= 2\pi \neq 0
\end{aligned}
$$

Not orthonormal, hence orthonormalize:

$$\varphi_1(x) = \frac{x}{[\int_{-\pi}^{\pi} x \cdot x dx]^{0.5}} = \sqrt{\frac{3}{2\pi^3}} x$$

$$\bar{\varphi}_2(x) = sin(x) - \varphi_1(x) \int_{-\pi}^{\pi} sin(x)\varphi_1(x)dx = sin(x) - \frac{3}{\pi^2}x$$

Normalise $\bar{\varphi}_2(x)$:

$$\varphi_2(x) = \frac{sin(x) - \frac{3}{\pi^2}x}{[\int_{-\pi}^{\pi}(sin(x) - \frac{3}{\pi^2}x) \cdot (sin(x) - \frac{3}{\pi^2}x)dx]^{0.5}} = \frac{sin(x) - \frac{3}{\pi^2}x}{\sqrt{\pi - \frac{6}{\pi}}}$$

b) What is the main reason to use orthonormal basis functions for interpolations?
Adding/removing of terms without the recomputation of coefficients.

Question 9: Backpropagation (10 points)

In this exercise, we are going to perform steps of the back-propagation method in a trivial neural network with a $\tanh$ activation function. The input is two dimensional $x = (x_1, x_2)^T$ while the output is one dimensional $o \in \mathbb{R}$. The neural network architecture is

$$o = \tanh(z), \quad z = w_1 x_1 + w_2 x_2 + b,$$

where $w_1, w_2, b \in \mathbb{R}$ are the weight parameters.

a) Prove that the derivative of the $\tanh$ activation function $f(x) = \tanh(x)$ takes the form

$$f'(x) = 1 - f^2(x),$$

using the fact that

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

$$\begin{aligned}
\tanh(x) &= \frac{e^x - e^{-x}}{e^x + e^{-x}} \implies \\
\partial_x \tanh(x) &= \frac{(e^x + e^{-x})\partial_x(e^x - e^{-x}) - (e^x - e^{-x})\partial_x(e^x + e^{-x})}{(e^x + e^{-x})^2} \\
&= \frac{(e^x + e^{-x})^2 - (e^x - e^{-x})^2}{(e^x + e^{-x})^2} \\
&= 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} \\
&= 1 - \tanh(x)^2.
\end{aligned}$$

b) Assume that you are given only one data-point $(x_1, x_2, \tilde{o}) = (1, 5, 1.5)$, where $\tilde{o}$ is the target value. The weights are initialized according to $(w_1, w_2, b) = (0.2, 0.07, 0)$. Compute the forward pass. Report values for z and o. Use the approximation $\tanh(0.55) \approx 0.5$.

$$z = w_1 x_1 + w_2 x_2 + b = 0.2 \cdot 1 + 0.07 \cdot 5 + 0 = 0.55$$
$$o = \tanh(0.55) = 0.5$$

c) The loss function is given as $\mathcal{L}(w_1, w_2, b) = (o - \tilde{o})^2$. Compute $\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial w}$ for $w \in \{w_1, w_2, b\}$. After deriving the three analytic expressions, replace values for the sample $(x_1, x_2, \tilde{o}) = (1, 5, 1.5)$.

$$\frac{\partial \mathcal{L}}{\partial z} = \frac{\partial (o - \tilde{o})^2}{\partial z} = \frac{\partial (\tanh(z) - \tilde{o})^2}{\partial z} =$$
$$= 2\Big(\tanh(z) - \tilde{o}\Big)\partial_z \tanh(z) = 2\Big(\tanh(z) - \tilde{o}\Big)\Big(1 - \tanh(z)^2\Big)$$

$$\frac{\partial z}{\partial w_1} = x_1, \quad \frac{\partial z}{\partial w_2} = x_2, \quad \frac{\partial z}{\partial b} = 1.$$

Replacing values for the sample $(x_1, x_2, \tilde{o}) = (1, 5, 1.5)$ we get

$$\frac{\partial \mathcal{L}}{\partial z} = 2\Big(\tanh(z) - \tilde{o}\Big)\Big(1 - \tanh(z)^2\Big) = 2 \cdot \Big(\tanh(0.55) - 1.5\Big)\Big(1 - \tanh(0.55)^2\Big) \implies$$
$$\frac{\partial \mathcal{L}}{\partial z} = 2 \cdot \Big(0.5 - 1.5\Big)\Big(1 - 0.25\Big) = 2 \cdot (-1) \cdot 0.75 = -1.5$$

$$\frac{\partial z}{\partial w_1} = x_1 = 1, \quad \frac{\partial z}{\partial w_2} = x_2 = 5, \quad \frac{\partial z}{\partial b} = 1.$$

Using these computed values we can finally get the desired partial derivatives as follows

$$\frac{\partial \mathcal{L}}{\partial w_1} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial w_1} = -1.5 \cdot 1 = -1.5$$
$$\frac{\partial \mathcal{L}}{\partial w_2} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial w_2} = -1.5 \cdot 5 = -7.5$$
$$\frac{\partial \mathcal{L}}{\partial b} = \frac{\partial \mathcal{L}}{\partial z}\frac{\partial z}{\partial b} = -1.5 \cdot 1 = -1.5$$

d) Use the error gradients computed in the previous question to perform one step of the gradient-descent algorithm with a step-size $\eta = 0.01$. The formula of gradient descent is

$$w^{i+1} = w^i - \eta \left.\frac{\partial \mathcal{L}}{\partial w}\right|_{w_1^i, w_2^i, b^i},$$

where i is the iteration number.

$$w_1^1 = w_1^0 - \eta \left.\frac{\partial L}{\partial w_1}\right|_{w_1^0, w_2^0, b^0} = 0.2 - 0.01 \cdot (-1.5) = 0.215$$
$$w_2^1 = w_2^0 - \eta \left.\frac{\partial L}{\partial w_2}\right|_{w_1^0, w_2^0, b^0} = 0.07 - 0.01 \cdot (-7.5) = 0.145$$
$$b^1 = b^0 - \eta \left.\frac{\partial L}{\partial b}\right|_{w_1^0, w_2^0, b^0} = 0 - 0.01 \cdot (-1.5) = 0.015$$

e) Did the loss decrease after the gradient-descent step? If yes, is this always the case? If not, what may be the cause of the increase? Use the approximation $\tanh(0.955) \approx 0.75$.

$$z = w_1x_1 + w_2x_2 + b = 0.215 \cdot 1 + 0.145 \cdot 5 + 0.015 = 0.955$$
$$o = \tanh(0.955) = 0.75$$
$$L = (o - \tilde{o})^2 = (0.75 - 1.5)^2 = 0.75^2 = 0.5625$$

The error before applying the first step of gradient descent was $L = (o - \tilde{o})^2 = (0.5 - 1.5)^2 = 1$. As a consequence, the error was reduced. This is not always the case, as gradient descent is not monotonically decreasing. A large learning rate may cause an increase of the error. If the student has arithmetic errors that caused the error to increase, the correct answer to the second question is: The cause is the large learning rate and should be reduced.

Question 10: Newton's method convergence (9 points)

Consider 4th order polynomials, i.e., $f(x) = a + bx + cx^2 + dx^3 + ex^4$, with $e = 1$. Newton's method is used to find the roots of the functions $f(x)$.

a) What is the order of convergence of Newton's method in the vicinity of simple roots?

Quadratic convergence.

b) Assume that α is a root of $f(x)$. Under what **condition** is the convergence order in the vicinity of the root $x = \alpha$ linear?

Linear if $f'(\alpha) = 0$.

c) Assume that $-\alpha$ is also a root of $f(x)$. Under what **conditions** is the convergence order in the vicinity of the root $x = -\alpha$ cubic?

Cubic if $f'(-\alpha) \neq 0$, $f''(-\alpha) = 0$.

d) How many roots does $f(x)$ have? Express the polynomial $f(x)$ in its root factored form $\prod_i (x - \beta_i)$ where β_i's are the roots. Substitute the roots you know.

Three roots. $f(x) = (x - \alpha)^2(x + \alpha)(x - \beta)$

e) Use the conditions you found in the previous sub-questions and express the remaining root(s) in terms of α.

The derivatives are:

$$\begin{aligned}
f'(x) &= 2(x-\alpha)(x+\alpha)(x-\beta) + (x-\alpha)^2(x-\beta) + (x-\alpha)^2(x+\alpha) \\
f''(x) &= 2(x+\alpha)(x-\beta) + 2(x-\alpha)(x-\beta) + 2(x-\alpha)(x+\alpha) \\
&\quad + 2(x-\alpha)(x-\beta) + (x-\alpha)^2 + 2(x-\alpha)(x+\alpha) + (x-\alpha)^2 \\
&= 2(x+\alpha)(x-\beta) + 4(x-\alpha)(x-\beta) + 4(x-\alpha)(x+\alpha) + 2(x-\alpha)^2,
\end{aligned}$$

$$f''(-\alpha) = 4(-2\alpha)(-\alpha-\beta) + 2(-2\alpha)^2 = 8\alpha(\alpha+\beta) + 8\alpha^2 = 16\alpha^2 + 8\alpha\beta = 0. \quad (1)$$

The unknown β is thus $\beta = -2\alpha$.

f) What is/are the convergence rate of Newton's method in the vicinity of the remaining root(s).

Since $\beta \neq \pm\alpha$ the root β is a simple root, thus has at least quadratic convergence.

$$f''(\beta) = 4(\beta - \alpha)(\beta + \alpha) + 2(\beta - \alpha)^2 = 4(-3\alpha)(-\alpha) + 2(-3\alpha)^2 = 30\alpha^2 \neq 0 \qquad (2)$$

The convergence is quadratic.

Question 11: Romberg Integration (6 points)

You are asked to compute the integral of the simplest form of the so called rational function equation $f(x) = \int_1^2 \frac{1}{x} dx$. You have forgotten that the formula can be directly integrated, its result being the $ln(2) = 0.69314718$ and you try to approximate the integral numerically. You are asked to:

a) Compute its numerical approximation of the integral using the trapezoidal rule on a **single** interval.

In order to compute the trapezoidal approximation of $f(x)$, we need to evaluate the function only at the end points of the integral, as follows:

$$I_{Trap} = R_0^1 = (2-1)\left(1 + \frac{1}{2}\right)\frac{1}{2} = 0.75$$

b) Proceed to a more accurate computation resorting to a Romberg integration scheme of order 1 and subsequently of order 2, i.e. R_1^1, R_2^1. In the process you build up a table in a form of a tree diagram to keep track of the Romberg computations.

To compute the Romberg integration result of order 1, we need to make use of the midpoint of the integral:

$$R_0^2 = \frac{1}{2}R_0^1 + \frac{1}{2}\frac{1}{1.5} = \frac{1}{2}0.75 + \frac{1}{3} = 0.70833333$$

and further subdiving the interval we obtain:

$$R_0^4 = \frac{1}{4}\left(\frac{1}{1.25} + \frac{1}{1.5} + \frac{1}{1.75}\right) + \frac{1}{8}(1 + 0.5) = 0.69702381$$

We can now compute the first order Romberg approximation, as follows:

$$R_1^1 = \frac{4 * R_0^2 - R_0^1}{4-1} = 0.69444444$$

upon which we can compute the second order Romberg approximation, after computing $R_1^2 = \frac{4 * R_0^4 - R_0^2}{4-1} = 0.693253$, as follows:

$$R_2^1 = \frac{16 * R_1^2 - R_1^1}{16-1} = 0.69317460$$

The tree diagram of Romberg approximations that we followed can be pictured as follows:

$$\begin{array}{lll} R_0^1 & & \\ R_0^2 & R_1^1 & \\ R_0^4 & R_1^2 & R_2^1 \end{array}$$

Question 12: Pseudocode (9 points)

a) The following pseudo-code is supposed to perform numerical integration based on the Simpson's rule of a function $f(x)$ on the interval $[a, b]$ using N subintervals. Find the errors.

Algorithm 1 Simpson's quadrature with bugs

Input:

a, b, {integration interval}
N, {number of subdivisions}

Output:

$\mathcal{I}$, {estimate of the integral}

Steps:

$h = (b - a)/N$
$\mathcal{I}_0 = f(a) + f(b)$
$\mathcal{I}_1 = 0.0$
$\mathcal{I}_2 = 0.0$
for $i \leftarrow 1, \ldots, N$ **do**
  $\tilde{x} = a + hi$
  if i even **then**
    $\mathcal{I}_2 = f(\tilde{x})$
  else
    $\mathcal{I}_1 = f(\tilde{x})$
  end if
  $\mathcal{I} = \mathcal{I} + h(\mathcal{I}_0 + 2\mathcal{I}_2 + 4\mathcal{I}_1)/3$
end for
return $\mathcal{I}$

The correct pseudo code is given in the following:

Steps:

$h = (b - a)/N$
$\mathcal{I}_0 = f(a) + f(b)$
$\mathcal{I}_1 = 0.0$
$\mathcal{I}_2 = 0.0$
for $i \leftarrow 1, \ldots, N - 1$ **do**
  $\tilde{x} = a + hi$
  if i even **then**
    $\mathcal{I}_2 = \mathcal{I}_2 + f(\tilde{x})$
  else
    $\mathcal{I}_1 = \mathcal{I}_1 + f(\tilde{x})$
  end if
end for
$\mathcal{I} = h(\mathcal{I}_0 + 2\mathcal{I}_2 + 4\mathcal{I}_1)/3$
return $\mathcal{I}$

b) The following pseudo-code is supposed to perform least squares regression to compute the coefficients $x \in \mathbb{R}^m$ of the problem $Ax = b$. Find the errors.

Algorithm 2 Least squares regression with bugs

Input:

A, {matrix $A \in \mathbb{R}^{n\times m}$}
b, {vector $b \in \mathbb{R}^n$}

Output:

x, $\{x \in \mathbb{R}^m\}$

Steps:

Compute the transpose of A, i.e. A^T
Compute $C = AA^T$
Compute C^{-1}
Compute $d = Ab$
Compute $x = C^{-1}d$.
return x

The correct pseudo code is given in the following:

Steps:

Given $A \in \mathbb{R}^{n\times m}$, $b \in \mathbb{R}^n$
Compute the transpose of A, i.e. A^T
Compute $C = A^T A$
Compute C^{-1}
Compute $d = A^T b$
Compute $x = C^{-1}d$.
return x

c) Assume you are given a **closed** surface in $\mathbb{R}^3$ described by the relationship $g(x, y, z) = 0$. The interior of this surface is given by $\{(x, y, z)|g(x, y, z) < 0\}$. You assume that your curve lies in the rectangular box $\{(x, y, z)\,|\,x_- \leq x \leq x_+,\ y_- \leq y \leq y_+,\ z_- \leq z \leq z_+\}$. The following pseudo-code is used to approximate the interior volume defined by the surface based on Monte Carlo. Can you spot the errors?

Algorithm 3 Monte Carlo with bugs

Input:

$x_-, x_+, y_-, y_+, z_-, z_+$, {boundaries of the rectangular box}

Output:

V, {approximation of the interior volume}

Steps:

Compute $|\Omega| = (x_+ - x_-)(y_+ - y_-)$
$K = 0$ # counter
for $i \leftarrow 1, \ldots, N$ **do**
  Sample $x_i \sim uniform(-1, 1)$
  Sample $y_i \sim uniform(-1, 1)$
  Sample $z_i \sim uniform(-1, 1)$
  Compute $\tilde{g} = g(x_i, y_i, z_i)$.
  if $\tilde{g} \geq 0$ **then**
    $K = K + 1$
  end if
end for
$V = K|\Omega|$
return V

The correct pseudo code is given in the following:

Steps:

Compute $|\Omega| = (x_+ - x_-)(y_+ - y_-)$ $(z_+ - z_-)$
$K = 0$ # counter
for $i \leftarrow 1, \ldots, N$ **do**
  Sample $x_i \sim uniform(x_-, x_+)$
  Sample $y_i \sim uniform(y_-, y_+)$
  Sample $z_i \sim uniform(z_-, z_+)$
  Compute $\tilde{g} = g(x_i, y_i, z_i)$.
  if $\tilde{g} < 0$ **then**
    $K = K + 1$
  end if
end for
$V = K/N|\Omega|$
return V

Good luck!