Engineering FIRST YEAR

Part IA Paper 4: Mathematical Methods

Examples Paper 1 – Computing

*Straightforward exercises are marked †, challenging problems are marked *.*
Answers can be found at the end of the paper.

– You may need to revise material in the Michaelmas Term Activity Notebooks[1] to complete this examples paper.

– For questions that involve computation you should implement your algorithms in Python. Under examination conditions, you would be asked to express algorithms in Python/Python-like pseudocode. The most important requirement of pseudocode is that it clearly describes an algorithm.

1. † For the following problems suggest an appropriate Python data structure:
   (a) A telephone directory that maps names to telephone numbers.
   (b) A list of motor vehicles in the UK that stores the registration, engine type and colour for each vehicle. If any of this data is changed for a vehicle it is considered to be a new vehicle.
   (c) Anonymised student scores on a test that will be used for statistical analysis. The cohort is large and the scores are not necessarily whole numbers.

2. Determine the minimum number of bits required to store the below integers, assuming that one bit is used to store the sign and following the convention that the only representation of zero is $0\,0 \ldots 0$.
   (a) 48
   (b) -48
   (c) 2500455245
   (d) -8

3. We wish to evaluate the function
$$\frac{1}{\sqrt{x^2+1}-x}$$
   for $x = 10^n$ where n is a positive integer, using floating point numbers (floats). Estimate the largest n that avoids division-by-zero for:
   (a) 32-bit floats (approximately 7 significant digit precision).
   (b) 64-bit floats (approximately 15 significant digit precision).

4. * For two real numbers x and y, the floating point representations $x^\star$ and $y^\star$ are:
$$x^\star = x\,(1+\epsilon_x)\,,$$
$$y^\star = y\,\big(1+\epsilon_y\big)\,,$$
   where $|\epsilon_x|$, $|\epsilon_y| \leq \epsilon$. The term ϵ represents the maximum relative error due to round-off[2], and is small ($\epsilon \ll 1$). The relative error is given by $|z - z^\star|/|z|$.

   Accounting for round-off errors, determine the maximum relative error in terms of ϵ for the following operations:

[1] `https://notebooks.azure.com/garth-wells/libraries/CUED-IA-Computing-Michaelmas`
[2] For a 64-bit float $\epsilon \approx 10^{-16}$. This is for interest only and is not required for completing the question.

(a) $10x$
(b) xy
(c) x/y
(d) $x+y$

Comment on the sensitivity of the relative error for each operation compared to ϵ.

5. Evaluate the minimum algorithmic complexity of the following operations:

   (a) $2\boldsymbol{u}+\boldsymbol{v}$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are vectors of length n.
   (b) $\boldsymbol{ABu}+\boldsymbol{Bu}$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are $n \times n$ matrices and $\boldsymbol{u}$ is a vector of length n.
   (c) Computing the mean of the entries of a vector of length n.
   (d) Computing the standard deviation of the entries of a $m \times n$ matrix.

6. † The integral of a function can be approximated by:

$$\int_a^b f \,\mathrm{d}x \approx (b-a)\sum_{i=0}^{n-1} w_i f(x_i),$$

   where w_i is a weight and x_i is a point between a and b. This is known as *numerical integration*.

   Develop a Python implementation of numerical integration that, given a function f, limits a and b, locations x_i and weights w_i, approximates the integral. Use your algorithm to approximate $\int_0^{10} x^3 + x^2 \,\mathrm{d}x$ using:

   (a) Trapezoidal rule: $n=2$, $w_i = 1/2$, $x_0 = a$, $x_1 = b$.
   (b) Simpson's rule: $n=3$, $w_0 = w_2 = 1/6$, $w_1 = 2/3$, $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$.
   (c) Two-point Gauss quadrature: $n=2$, $w_i = 1/2$, $x_i = (1/2)(a+b \pm (b-a)/(\sqrt{3}))$.

   Compute the error for each case by comparing to the exact solution.

   For a more complex function, how could you modify your algorithm to improve the accuracy when using one of the above schemes?

7. * A mass–spring problem with friction is modelled by the ordinary differential equation

$$m\ddot{x} + kx = F,$$

   where x is the displacement, m is the mass, k is the spring stiffness and F is the friction force. The friction force depends on the direction of motion, and is given by:

$$F = \begin{cases} f & \dot{x} < 0, \\ -f & \dot{x} > 0, \\ 0 & \dot{x} = 0, \end{cases}$$

   where $f \geq 0$ is a constant.

   At time t_n, where n is an integer, we can write

$$m\ddot{x}_n + kx_n = F_n. \tag{1}$$

   The notation 'x_n' means x evaluated at time t_n, i.e. $x(t_n)$. The second derivative, $\ddot{x}_n$, can be approximated by:

$$\ddot{x}_n \approx \frac{x_{n-1} - 2x_n + x_{n+1}}{\Delta t^2}, \tag{2}$$

   where $\Delta t = t_{n+1} - t_n$ (assume constant).

(a) Derive the analytical solution to eq. (1) for the case $f = 0$ and the initial conditions $x(0) = 0.01$ and $\dot{x}(0) = 0$.

(b) Combine eq. (1) and eq. (2) to express x_{n+1} as a function of F_n, x_n and x_{n-1}. Use this expression to compute a numerical solution from $t = 0$ to $t = 20$ using $\Delta t = 0.01$, $m = 1$, $k = 40$, $f = 0$, and setting $x_0 = x_1 = 0.01$ (this is equivalent to the initial condition in item 7a). On a single graph, plot the numerical and analytical solutions.

(c) For $f = 0.025$, compute the numerical solution and plot on the same graph as the analytical solution for the $f = 0$ case from item 7a.
Hint: approximate the velocity by $v_n = (x_n - x_{n-1})/\Delta t$

8. The mean of a function $f(x, y)$ over a rectangle $(-a, a) \times (-b, b)$ is given by:

$$\bar{f} = \frac{1}{4ab} \int_{-b}^{b} \int_{-a}^{a} f(x, y) \, \mathrm{d}x \, \mathrm{d}y,$$

hence if we know the mean of the function over the integration domain and the area of the domain, we can determine the integral. (In simple terms, a double integral is the 'volume under the surface', hence $\bar{f}$ is the 'mean height' of f. Discuss with your supervisor if the concept is unfamiliar.)

We can approximate the mean of f by

$$\bar{f} \approx \frac{1}{N} \sum_{i=0}^{N-1} f(x_i, y_i),$$

where x_i is a random value between $-a$ and a, and y_i is a random value between $-b$ and b (uniform distribution). Using this approximation of $\bar{f}$ to approximate the integral of f is known as *Monte Carlo integration*.

Design an implementation of Monte Carlo integration for a function $f(x, y)$, and use it to:

(a) Approximate $\int_{-1}^{1} \int_{-1}^{1} e^{xy} \cos^2(y) \sin(x^2) \, \mathrm{d}x \, \mathrm{d}y$.

(b) * Approximate π by approximately integrating a suitable function over the domain $(-1, 1) \times (-1, 1)$.

Use $N = 10^2$, 10^5 and 10^6 in your tests.

Answers

1. (a) `dictionary` (b) `list` of `tuple` (c) NumPy array
2. (a) 7 (b) 7 (c) 33 (d) 4
3. (a) 3 (b) 7
4. (a) ϵ (b) 2ϵ (c) 2ϵ (d) $\epsilon(|x| + |y|)/|x + y|$
5. (a) $O(n)$ (b) $O(n^2)$ (c) $O(n)$ (d) $O(mn)$
6. (a) Error: 2666.67 (b) Error: 0 (c) Error: 0
7. –
8. –

GNW
Michaelmas 2018