Optimisation and matrices in MATLAB

The MATLAB function `fminsearch` can be used to numerically approximate local minimisers. As arguments, it requires a function to be minimised and an initial point to start with. However, the input function needs a specific form: it **must** take a single argument, which is assumed to be an array containing the components of the input vector. If the function is defined in a function file, to pass it to `fminsearch`, its name must be referenced by using an at-sign (`@`). Here is an example that will find a minimiser for Himmelblau's function,

$$f(\mathbf{x}) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.$$

Define the following function in a file `himmelblau.m`:

```
function f = himmelblau(x)
f = (x(1)^2 + x(2) - 11)^2 + (x(1) + x(2)^2 - 7)^2;
end
```

Then by running the following, a local minimiser can be found:

```
fminsearch(@himmelblau, [0 0])
```

Compare this to the approach given Workshop 1, where an anonymous function was used instead of a function file; in that case, the at-sign was not needed.

1. In a function file `rosenbrock.m`, implement Rosenbrock's function (see Workshop 2) so that it is compatible with `fminsearch`. Then, use `fminsearch` to find the minimiser of Rosenbrock's function.

Moreover, when plotting a function of one variable, MATLAB's internal mechanisms require that the function can be evaluated elementwise. For example, evaluating `f([1 2 3])` must result in the list `[f(1) f(2) f(3)]`. Most everyday functions act elementwise without any caveats, but for a few special cases, there are special commands to use. For example, since the `^` operation is reserved for matrix powers, to square every element in an array requires using `.^` instead. For example:

```
>> x = [1 2 3];
>> x.^2
ans =
     1     4     9
```

The next two questions will acquaint you with elementwise operations, matrix indexing and plotting in MATLAB.

2. Perform each of the following **using a single line of code**, updating the matrix `M` as you go.

   (a) Create a matrix `M` with the numbers 2, 5, 8, ..., 50 in the first row and 101, 98, ..., 53 in the second row.

   (b) Add 1 to every element in `M` and then divide every element by 3.

   (c) Delete the first two columns of `M`.

   (d) Assign the number of rows and number of columns of `M` to the variables `nrow` and `ncol`.

   (e) Delete a random column of `M` (using the `randi` function).

   (f) Swap both rows of `M`.

   (g) Swap columns 7 and 9 of `M`.

   (h) Apply the map $x \to (x^2 - x)/2$ to the elements in `M`.

   (i) Replace each element in the second row of `M` with a random number between -7 and 7.

   (j) Replace each element in columns 3–6 with a random number between -10 and 10.

   (k) Replace all positive elements of `M` with their square root, rounded down to the nearest integer. *You will need the* `floor` *function.*

   (l) Replace all negative elements of `M` with the result of applying the map $x \mapsto (x^2 - x)/2$ to those elements.

3. Plot the function

$$f(x) = (2x^2 - 2x - 4) \cdot \log(x + 1)$$

   using 1000 subdivisions of the interval $[0, 3]$. Use the `xlabel`, `ylabel` and `title` commands to appropriately label each axis and give the graph a title. Then use `fminsearch` to find a local minimiser of f.

Systems of linear equations

Performing linear regression ultimately requires solving systems of linear equations. Practice solving systems of linear with Questions 4 to 8.

4. Solve the following system of linear equations by applying a single matrix multiplication:

$$\begin{aligned} 3y - x &= 4, \\ 3x - 5y &= 2. \end{aligned}$$

5. Perform Gaussian elimination to solve the system of equations in Question 4.

6. Perform Gaussian elimination to solve the following system of equations:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + x_4 &= 0, \\ x_2 + 2x_3 + 3x_4 &= 0, \\ 2x_1 + x_2 &= 0. \end{aligned}$$

7. Use MATLAB commands to perform the same matrix product and row operations you did in Questions 4–6.

8. Check your answers to Questions 5 and 6 using MATLAB's `rref` function.

Linear regression

Let's now investigate least squares regression for linear functions. The purpose of these questions is to understand how the objective function is obtained, how the normal equations are obtained, and how they can be solved to find the values of the coefficients.

9. Consider the following set of data:

i	1	2	3	4
x_i	-4	-2	2	5
y_i	-1	0	2	3

Your aim is to fit these data points to a linear function of the form $f(x) = a_0 + a_1 x$ using least squares regression.

(a) Write down the objective function that needs minimising.

(b) Implement the objective function of (a) in MATLAB and then use the `fminsearch` function to approximate the coefficients.

(c) Apply the FONC directly to obtain a system of two linear equations that can be solved to find the coefficients a_0 and a_1. Solve this system of equations by applying a single matrix multiplication.

(d) Use the general normal equations from Section 2.1 of the reading notes to write down the system of two linear equations that can be solved to find the coefficients. Express the system in matrix form, and use MATLAB to solve it.

(e) Write down matrices $\mathbf{A}$, $\mathbf{a}$ and $\mathbf{y}$ for which solving $\mathbf{A}^T\mathbf{A}\mathbf{a} = \mathbf{A}^T\mathbf{y}$ finds the coefficients. Implement them in MATLAB and solve the system of equations to find the coefficients.

10. Suppose you had a set of n data points of the form (x_i, y_i, z_i), and wished to use the method of least squares to fit these data to a linear function of the form

$$z = f(x, y) = ax + by + c.$$

(a) Write down the objective function that needs minimisation.

(b) Use the FONC to find a system of linear equations that can be solved to find the coefficients.

(c) Express this system of equations in matrix form.

Reading data in MATLAB

In practice, data is often supplied in CSV (comma-separated values) format, which is a file format that holds a plain-text list of data. Typically, each row of the CSV represents a data entry, and each column corresponds to a variable or attribute. There is a CSV file on the LMS called `workshop3_data.csv` with 15 entries, and columns for variables `x`, `y` and `z`.

11. You will be fitting the data in `workshop3_data.csv` to a linear model of the form

$$y = a_0 + a_1 x,$$

using least squares regression.

(a) Save the file in the current MATLAB folder shown on the left of your MATLAB window. Then import the data into MATLAB using the command `dat = readmatrix('workshop3_data')`.

(b) Open the CSV file in Excel (or your preferred spreadsheet software) and look at the first row. Extract the first row of the matrix `dat`. Does it equal the first row of the file?

(c) Define variables `x`, `y` and `z` which are, respectively, the first, second and third columns of `dat`.

(d) Use the `scatter` function to create a scatter plot with `x` on the horizontal axis and `z` on the vertical axis.

(e) Define suitable matrices and apply suitable operations in MATLAB to find the coefficients that best fit the linear model $z = a_0 + a_1 x$.

(f) In the same figure as part (d), plot the line $a_0 + a_1 x$.

12. Using your answer to Question 10(c), define suitable matrices and apply suitable operations in MATLAB to find the coefficients that best fit the linear model $z = a_0 + a_1 x + a_2 y$ to the data in `workshop3_data.csv`.