Semester 2, 2020/21
University of Surrey

Lewis Napper
Stefan Klus

Numerical and Computational Methods
Exercise Solutions

Contents

This file contains the solutions to the mathematical problems from exercise sheets $1-5$. The MATLAB scripts for exercise sheet n can be found in *"Exercises_n_Solutions_Script.m"*. The code block *"Class n Exercise m Script"* can be executed to view the output of the functions for a given exercise m on sheet n. The functions themselves can be found in the block *"Class n Exercise m Function(s)"*.

Exercise Sheet 1

Exercise 1.1.

For the numbers a given in the question, computing the cube root of a to an accuracy of 10^{-6} takes the following amount of iterations:

(i) For $a = 3.375$, we need 6 iterations.

(ii) For $a = -8$, we need 7 iterations.

(iii) For $a = 1331$, we need 16 iterations.

Exercise 1.2.

This exercise is purely numerical.
See *"Exercises_1_Solutions_Script.m"* for the corresponding code.

Exercise 1.3.

The algorithm approximates the largest eigenvalue of the matrix A.

Exercise 1.4.

When timing the Laplace expansion against Matlab's inbuilt functions, the key points are

- The Laplace expansion takes a lot longer to compute for large matrix sizes. Recall the algorithm has time complexity $O(n!)$.
- The inbuilt function is consistently faster than the Laplace expansion. The inbuilt function uses LU decomposition which has time complexity $O(n^3)$).

Exercise Sheet 2

Exercise 2.1.

This exercise is purely numerical.
See *"Exercises_2_Solutions_Script.m"* for the corresponding code (including functions for LU decomposition, forward substitution, and backward substitution).

Exercise 2.2.

This exercise is predominantly numerical, however we make a few key observations:

- By construction, $b = H * (1,1,1,1,1)^T$, so $x = (1,1,1,1,1)^T$ is the solution for the Hilbert problem.
- Perturbing the vector b by some small vector, e.g. $(10^{-3})e_1$, results in a wildly different answer. In this case we find

$$x = \begin{pmatrix} 1.0250 \\ 0.7000 \\ 2.0500 \\ -0.4000 \\ 1.6300 \end{pmatrix}. \tag{2.1}$$

- For our choice of perturbation, we find that the condition number is $c \approx 4.8 * 10^5$. Since the condition number is large, we can see that Hilbert matrices are ill-conditioned, which explains the behaviour above.
- The runtimes of the LU Solver algorithm applied to the Hilbert problem and its perturbation are roughly the same, with spikes being amplified slightly by the perturbation.

Exercise 2.3.

Let A be a symmetric, positive definite, $n \times n$ matrix. This means that $x^T Ax > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$.

(i) First note that a symmetric matrix has only real eigenvalues. Let v be an arbitrary eigenvector of A. If all eigenvalues of A are positive, then

$Av = \lambda v$ for some eigenvalue $\lambda > 0$. As A is symmetric and positive definite, we have

$$0 < v^T A v = \lambda v^T v = \lambda ||v||^2 \,. \tag{2.2}$$

Since $||v||^2 > 0$, we find $\lambda > 0$, as required.

(ii) Recall that symmetric, positive definite matrices admit the following diagonalisation/eigendecomposition:

$$A = VDV^T \qquad \text{with} \qquad D = \mathrm{diag}(\lambda_1, \cdots \lambda_n) \tag{2.3}$$

the diagonal matrix whose entries are the eigenvalues of A. Here V is the matrix formed by taking the eigenvectors of A as columns. A $V^T = V^{-1}$ by the real spectral theorem (the eigenvector matrix is orthogonal), we can apply Binet's theorem to find

$$\det(A) = \det(D) = \prod_{i=1}^{n} \lambda_i > 0 \,. \tag{2.4}$$

(iii) As $\det(A) \neq 0$, the matrix A must be invertible.

(iv) Again using the eigendecomposition, we find

$$A^{-1} = (VDV^T)^{-1} = VD^{-1}V^T \,. \tag{2.5}$$

Evidently then, A^{-1} is symmetric. Furthermore, given any vector $x \in \mathbb{R} \setminus \{0\}$, we have

$$\begin{aligned} x^T A^{-1} x &= x^T V D^{-1} V^T x && (2.6)\\ &= x^T V D^{-\frac{1}{2}} D^{-\frac{1}{2}} V^T x && (2.7)\\ &= ||D^{-\frac{1}{2}} V^T x||^2 > 0 \,, && (2.8) \end{aligned}$$

so A^{-1} is positive definite, as required.

Exercise Sheet 3

Exercise 3.1. (i) Recall that a function $F : \mathbb{R} \to \mathbb{R}$ is called a contraction if there exists a Lipschitz constant $K < 1$ such that

$$|F(x) - F(y)| \leq K|x - y| \,, \tag{3.1}$$

for all $x, y \in \mathbb{R}$. By the process described in the hint, if the derivative of F is bounded by some $M < 1$, i.e. $|F'(x)| < M < 1$, then $K = M$ is an appropriate choice of Lipschitz constant. It remains to check that $|F'(x)|$ is bounded above by such an M in our case, as shown by the plot of the function dF.

(ii) Using the plot of the error against the number of iterations, we see exponential decay in the error as the number of iterations is increased. This is mirrored by the ratio of successive errors, which converges to a fixed value, suggesting the scheme itself converges.

(iii) We find the polynomial by setting $F(x) = x$ (as we consider fixed point iteration) and multiplying out the denominators to obtain:

$$x^3 - x^2 - x - 1 = 0 \,. \tag{3.2}$$

Exercise 3.2.

This is a predominantly numerical exercise.
See *"Exercises_3_Solutions_Script.m"* for the corresponding code. The root of $f(x) = x - 1 - \frac{1}{x} - \frac{1}{x^2}$ is in the interval $[1.8391, 1.8394]$ according to the bisection method.

Exercise 3.3.

We are given three points $(x_0, y_0) = (-2, 5)$, $(x_1, y_1) = (0, -1)$, and $(x_2, y_2) = (3, 8)$. Recall that the Lagrange polynomials are given by

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \,, \tag{3.3}$$

for $i = 0, \cdots, n$ (here $n = 2$), from which we find

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{1}{10}x^2 - \frac{3}{10}x \,, \tag{3.4}$$

$$L_1(x) = \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} = -\frac{1}{6}x^2 + \frac{1}{6}x + 1\,, \tag{3.5}$$

$$L_2(x) = \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} = \frac{1}{15}x^2 + \frac{2}{15}x\,, \tag{3.6}$$

and the interpolating polynomial through the points (x_0, y_0), (x_1, y_1), and (x_2, y_2) is given by

$$p(x) = \sum_{i=0}^{2} y_i L_i(x) = \frac{6}{5}x^2 - \frac{3}{5}x - 1\,. \tag{3.7}$$

It is straightforward to verify that $p(-2) = 5$, $p(0) = -1$, and $p(3) = 8$, so the polynomial passes through the given points, as required. We also compute the Lagrange polynomials and interpolating polynomial in MATLAB and obtain the same result numerically.

Exercise 3.4.

Given three points $(x_0, y_0) = (-2, 5)$, $(x_1, y_1) = (0, -1)$, and $(x_2, y_2) = (3, 8)$, the Newton interpolation formula for the interpolating polynomial of order two is

$$p(x) = p(x_0) + p[x_0, x_1](x - x_0) + p[x_0, x_1, x_2](x - x_0)(x - x_1)\,, \tag{3.8}$$

where $p(x_0) = y_0$ and the divided differences are given by

$$p[x_i, x_j, x_k] = \frac{p[x_k, x_j] - p[x_j, x_i]}{x_k - x_i} \qquad \text{for } i \neq k\,, \tag{3.9}$$

$$p[x_i, x_j] = \frac{y_j - y_i}{x_j - x_i} \qquad \text{for } i \neq j\,. \tag{3.10}$$

Computing these differences for our points, we find

$$p[x_0, x_1] = -3\,, \quad p[x_1, x_2] = 3\,, \quad \text{and} \quad p[x_1, x_2, x_3] = \frac{6}{5} \tag{3.11}$$

so the interpolating polynomial is

$$p(x) = 5 - 3(x+2) + \frac{6}{5}(x+2)x = \frac{6}{5}x^2 - \frac{3}{5}x - 1 \tag{3.12}$$

as obtained in the previous part.

Exercise Sheet 4

Exercise 4.1.

We are given the following information

$$\begin{aligned}
&x_0 = 1, \quad x_1 = 2 && (4.1)\\
&y_0 = p(1) = 10, \quad y_1 = p(2) = 26 && (4.2)\\
&p'(2) = 23 && (4.3)\\
&p''(2) = 16\,. && (4.4)
\end{aligned}$$

Using Newton interpolation, the first two pieces of information would yield a linear interpolation p. However, as we are given derivative information, we can use Hermite interpolation. The Hermite interpolation formula for a third order interpolating polynomial given two points is

$$\begin{aligned}p(x) = p(x_0) + p[x_0, x_1](x - x_0) + p[x_0, x_1, x_1](x - x_0)(x - x_1)\\ + p[x_0, x_1, x_1, x_1](x - x_0)(x - x_1)^2\end{aligned}, \tag{4.5}$$

where, as in exercise 3.4, we have the divided differences

$$\begin{aligned}
&p[x_i, x_j, x_k, x_\ell] = \frac{p[x_j, x_k, x_\ell] - p[x_i, x_j, x_k]}{x_\ell - x_i} \quad \text{for } i \neq \ell\,, && (4.6)\\
&p[x_i, x_j, x_k] = \frac{p[x_k, x_j] - p[x_j, x_i]}{x_k - x_i} \quad \text{for } i \neq k\,, && (4.7)\\
&p[x_i, x_j] = \frac{y_j - y_i}{x_j - x_i} \quad \text{for } i \neq j\,. && (4.8)
\end{aligned}$$

When all n indices in a divided difference coincide, we also have the following formula

$$p[x_i, \cdots x_i] := \frac{1}{n!} p^{(n)}(x_i)\,. \tag{4.9}$$

Computing these differences for our points, we have

$$\begin{aligned}
&p[x_0, x_1] = 16\,, && (4.10)\\
&p[x_1, x_1] = p'(2) = 23\,, && (4.11)\\
&p[x_0, x_1, x_1] = \frac{p[x_0, x_1] - p[x_1, x_1]}{x_0 - x_1} = 7\,, && (4.12)\\
&p[x_1, x_1, x_1] = \frac{1}{2} p''(2) = 8\,, && (4.13)\\
&p[x_0, x_1, x_1, x_1] = 1\,. && (4.14)
\end{aligned}$$

Putting this all together, we find

$$p(x) = x^3 + 2x^2 + 3x + 4\,. \tag{4.15}$$

Upon computing the derivatives

$$p'(x) = 3x^2 + 4x + 3, \quad \text{and } p''(x) = 6x + 4\,, \tag{4.16}$$

it is straightforward to verify that $p(1) = 10$, $p(2) = 26$, $p'(2) = 23$, and $p''(2) = 16$, so our polynomial passes through the given points, as required.

Exercise 4.2.

This exercise is predominantly numerical, however we make one comment on why solving $C\alpha = r$ gives us an approximation of f. First recall that

$$C_{ij} = \langle \phi_i, \phi_j \rangle \quad \text{and} \quad r_i = \langle \phi_i, f \rangle\,, \tag{4.17}$$

where $\phi_i(x)$ for $i = 0, \cdots n$ is the basis against which we expand. Assume our function can be expanded with respect to this basis (this is true for the Taylor and Fourier bases), such that

$$f(x) = \sum_{i=0}^{n} = \alpha_i \phi_i(x)\,. \tag{4.18}$$

Therefore,

$$r_i = \langle \phi_i, \sum_{j=0}^{n} \alpha_j \phi_j \rangle = \sum_{j=0}^{n} \alpha_j \langle \phi_i, \phi_j \rangle = \sum_{j=0}^{n} \alpha_j C_{ij} = (C\alpha)_i\,. \tag{4.19}$$

That is, α in the equation $C\alpha = r$ is the vector of coefficients in the expansion of f against the basis $\phi_i(x)$.

Exercise 4.3.

This is a predominantly numerical exercise.
See *"Exercises_4_Solutions_Script.m"* for the corresponding code. Note that Simpson's rule converges as we increase the step size, with large gains initially but diminishing returns. The error of the method decreases with the step size.

Exercise Sheet 5

Exercise 5.1.

This question is predominantly numerical, however we show here how to obtain the analytical solutions to the given initial value problems.

(i) We are given the ODE $\frac{\mathrm{d}x}{\mathrm{d}t} = -2tx^2$ with initial condition $x(0) = 1$. Using separation of variables, we find

$$\frac{1}{x^2}\mathrm{d}x = -2t\,\mathrm{d}t\,. \tag{5.1}$$

Integrating both sides, we obtain

$$\frac{1}{x} = t^2 + C\,, \tag{5.2}$$

for some real constant C. Using our initial condition, we obtain $C = 1$, so

$$x = \frac{1}{t^2+1}\,. \tag{5.3}$$

A plot of this analytical solution against the approximation obtained via the standard Runge–Kutta method can be found in *"Exercises_5_Solutions_Script.m"* and this shows that the approximation is visually indistinguishable from the analytical solution.

(ii) The second ODE we consider is a system

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} \quad \text{with} \quad \begin{pmatrix} x(0) \\ y(0) \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}\,. \tag{5.4}$$

We solve this system by first finding the eigenvalues and eigenvectors of the matrix

$$M = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}\,. \tag{5.5}$$

The characteristic equation is $\lambda^1 + 1 = 0$, so the eigenvalues are $\pm\mathrm{i}$ and the eigenvectors satisfy

$$\begin{pmatrix} -\lambda & 1 \\ -1 & -\lambda \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \mp\mathrm{i} & 1 \\ -1 & \mp\mathrm{i} \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{5.6}$$

so we can take

$$v_\pm = \begin{pmatrix} \mp\mathrm{i} \\ 1 \end{pmatrix}\,. \tag{5.7}$$

Consequently,

$$\begin{pmatrix} x \\ y \end{pmatrix} = A\mathrm{e}^{\mathrm{i}t}\begin{pmatrix} -\mathrm{i} \\ 1 \end{pmatrix} + B\mathrm{e}^{-\mathrm{i}t}\begin{pmatrix} \mathrm{i} \\ 1 \end{pmatrix}\,. \tag{5.8}$$

Solving for A and B using our initial conditions, we find $A = -B = \frac{\mathrm{i}}{2}$, so

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{2}\begin{pmatrix} \mathrm{e}^{\mathrm{i}t} + \mathrm{e}^{-\mathrm{i}t} \\ \mathrm{i}(\mathrm{e}^{\mathrm{i}t} - \mathrm{e}^{-\mathrm{i}t}) \end{pmatrix} = \begin{pmatrix} \cos(t) \\ \sin(t) \end{pmatrix} . \tag{5.9}$$

Therefore, $x^2 + y^2 = \cos^2(t) + \sin^2(t) = 1$ and the analytic solution forms a circle in the (x, y)-plane, which is precisely what we see when performing the Runge–Kutta method.

Exercise 5.2.

The Butcher tableau for the four stage Runge–Kutta method in question is given by

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & & \frac{1}{2} & & \\ 1 & & & 1 & \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

From this we find

$$a = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} , \quad b = \begin{pmatrix} \frac{1}{6} \\ \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{6} \end{pmatrix} , \quad \text{and} \quad c = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \\ 1 \end{pmatrix} \tag{5.10}$$

where

$$k_i = f(t_0 + c_i h, x_0 + h \sum_{j=1}^{4} k_j) , \tag{5.11}$$

$$x_1 = x_0 + h \sum_{i=1}^{4} b_i k_i . \tag{5.12}$$

and $f(t, x) = Ax$ as per our ODE $x' = Ax$. Computing the k_i, we find

$$k_1 = f(t_0, x_0) = Ax_0 \tag{5.13}$$

$$k_2 = f(t_0 + \frac{1}{2}h, x_0 + \frac{1}{2}hk_1) = (1 + \frac{1}{2}hA)Ax_0 \tag{5.14}$$

$$k_3 = f(t_0 + \frac{1}{2}h, x_0 + \frac{1}{2}hk_2) = (1 + \frac{1}{2}hA + \frac{1}{4}h^2A^2)Ax_0 \tag{5.15}$$

$$k_4 = f(t_0 + h, x_0 + hk_3) = (1 + hA + \frac{1}{2}h^2A^2 + \frac{1}{4}h^3A^3)Ax_0 \tag{5.16}$$

hence

$$x_1 = x_0 + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4) \tag{5.17}$$

$$= (I + hA + \frac{1}{2}h^2A^2 + \frac{1}{6}h^3A^3 + \frac{1}{12}h^4A^4)x_0\,. \tag{5.18}$$

Consequently, our method is of the form $x_1 = Bx_0$ with

$$B = (I + hA + \frac{1}{2}h^2A^2 + \frac{1}{6}h^3A^3 + \frac{1}{12}h^4A^4)\,. \tag{5.19}$$

Note that the exact solution of the ODE $x' = Ax$ is

$$x(t) = e^{At}x_0 = (I + tA + \frac{1}{2}t^2A^2 + \frac{1}{6}t^3A^3 + \frac{1}{12}t^4A^4 + \cdots)x_0\,, \tag{5.20}$$

so our method produces a fourth order approximation to the exact solution at time $t = h$ (given initial condition $x(t = 0) = x_0$).

Exercise 5.3.

This question is predominantly numerical, once we make the observation that, for $|h\lambda| < 1$, the higher order terms in the exact solution become negligible and the solution is well approximated by B. The region of stability is plotted in *"Exercises_5_Solutions_Script.m"*.