

Please write your family and given names and **underline** your family name on the front page of your paper.

All questions **must** be written in latex, and compiled to a single pdf. Plots (7 for Q1 and 2 for Q3) must be embedded in latex. All code and output of Q1, Q3 and Q4 should be *embedded* into latex. Code and output should be embedded with fixed-width fonts, e.g. Courier. Font size of all fonts must be 12. Linespacing set to 1.1.

What to submit:

- (1) The single pdf file 00A3.pdf (with embedded code and output).
- (2) The latex source file of A3.
- (3) Any source code you wrote (for computation, plots, etc).
- (4) Plot files (7 for Q1 and 2 for Q3) in eps and png format.

Thus, the code and plots will be available within latex/pdf, *as well as* separately.

1. This question involves the study of the outbreak of an epidemic in a population and its eradication with vaccinations. The study of spreading of infectious diseases is mainly done by numerical Ordinary Differential Equations (ODEs), which are not the topic of CSC336 (but of CSC436). However, numerical ODE solvers, essentially convert a system of ODEs to a sequence of nonlinear systems. Each nonlinear system of the sequence relates unknown quantities at a time point (or step), in terms of known quantities at the previous time point, while the initial state is given (known). In this way, the values of various quantities at several time points are computed and studied.

Assume there is a population of N people, which remains constant over time. Assume that each person belongs to one of the five groups: susceptible, exposed (to become infected after incubation), infected/ious, recovered/immune and vaccinated/immune. The infected/ious persons transmit the virus at a given rate β to the susceptible and make them exposed, the exposed become infected/ious after a certain incubation period α^{-1} (also given), the infected/ious recover at a given rate γ and stay immune thereafter. Furthermore, the susceptible get vaccinated at a given rate ρ , and become immune immediately after vaccination and stay immune thereafter. The population of N persons is replenished at a given death and birth rate μ , and all newborns are susceptible.

In an instance of this problem, at each time point i , the following 5×5 nonlinear system needs to be solved:

$$x_1^{(i)} = x_1^{(i-1)} + h\mu N - h\mu x_1^{(i)} - h \frac{\beta}{N} x_1^{(i)} x_3^{(i)} - h\rho x_1^{(i)} \quad (1a)$$

$$x_2^{(i)} = x_2^{(i-1)} - h\alpha x_2^{(i)} - h\mu x_2^{(i)} + h \frac{\beta}{N} x_1^{(i)} x_3^{(i)} \quad (1b)$$

$$x_3^{(i)} = x_3^{(i-1)} - h\gamma x_3^{(i)} - h\mu x_3^{(i)} + h\alpha x_2^{(i)} \quad (1c)$$

$$x_4^{(i)} = x_4^{(i-1)} + h\gamma x_3^{(i)} - h\mu x_4^{(i)} \quad (1d)$$

$$x_5^{(i)} = x_5^{(i-1)} + h\rho x_1^{(i)} - h\mu x_5^{(i)}. \quad (1e)$$

In the above, $\bar{x}^{(i)} = [x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}, x_5^{(i)}]^T$ represents the vector of numbers of susceptible, exposed, infected/ious, recovered/resistant/immune and vaccinated/immune persons, respectively, at time point i , and these are the five unknowns that need to be computed at the i th time point (step), assuming the quantities of the previous time point $i-1$ are already computed, and the initial state, $\bar{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, x_3^{(0)}, x_4^{(0)}, x_5^{(0)}]^T$ is given. While in practice $x_j^{(i)}$ are integers, we treat them as continuous variables. Also, h is a small stepsize in time, that, in this instance, is given and fixed. Also note that, although some simplifications can be applied to this system, we just treat it as a general 5×5 nonlinear system. Finally, note that it is not required that you understand the details of how this nonlinear system arises. You take the system as granted.

- (a) [5 points] Write the system (1a)-(1e) in standard form $\bar{f}(\bar{x}^{(i)}) = \bar{0}$. Indicate all components of \bar{f} . Write (by hand, i.e. in latex) the Jacobian matrix for this system. Indicate all entries (some in terms of $x_j^{(i)}$).
- (b) [8 points] Consider Newton's method for this system. One iteration of Newton's includes computation of the form solve $J(\bar{x}^{(i,k-1)})\bar{s}^{(i,k-1)} = -\bar{f}(\bar{x}^{(i,k-1)})$
compute $\bar{x}^{(i,k)} = \bar{x}^{(i,k-1)} + \bar{s}^{(i,k-1)}$

Note that i is the time point index, while k is the index for Newton's iteration, the former remaining constant through all Newton iterations (all k).

Write a matlab or equivalent script that, given $N = 38e6$, $\alpha^{-1} = 6$, $\beta = 0.25$, $\gamma = 0.06$, $\mu = 0.01/365$, $\rho = 300\mu$, $h = 1/2$, $x_1^{(0)} = N - x_2^{(0)} - x_3^{(0)} - x_4^{(0)}$, $x_2^{(0)} = 20e3$, $x_3^{(0)} = 30e3$, $x_4^{(0)} = 850e3$, and $x_5^{(0)} = 0$, computes $\bar{x}^{(1)}$ by Newton's method with tolerance 10^{-6} . Use `maxit = 10`, and as stopping criterion the infinity norm of the residual vector. The Jacobian must be solved using `backslash`. At each Newton iteration, output $\bar{x}^{(1,k)}$ (the five values computed), the sum of the five values and the infinity norm of the residual. See script `testn1.m` (in the course website) for some template. Comment on the number of Newton iterations and on how the residual behaves as the iterations proceed.

- (c) [7 points] Consider now a simulation time period from 0 to $t_{end} = 150$ (can be viewed as a period of 150 days), divided in $nstep = \frac{t_{end}}{h}$ periods of stepsize (length) $h = 1/24$ (or dt), and that, at each time point i , given the state at time point $i - 1$, a nonlinear system of the form (1a)-(1e) is solved by Newton's method with tolerance 10^{-6} . Write a script that, given the same parameters as in (b), except that $\rho = 0$ (and $h = 1/24$), computes $\bar{x}^{(i)}$, for $i = 1, \dots, nstep$. Do NOT output and do NOT save the results of each Newton iteration. Save, but do NOT output, the number of Newton iterations at each timestep. Save, but do NOT output, the results ($\bar{x}^{(i)}$) from each time point. They will be used later for plotting. Output the initial values $\bar{x}^{(0)}$ and their sum, the computed final $\bar{x}^{(nstep)}$ (five values at time t_{end}), and their sum, as well as the maximum number of each type of persons among all time points (steps). Output also which **day** the maximum number of infected persons occurs.
- Plot $x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}$ and $x_5^{(i)}$ versus $ih, i = 0, \dots, nstep$ (index of time point scaled by h), in one plot, using solid (red), dashed (green), dashed-dotted (blue), dotted (black), and plain dot (cyan, 'c.') lines. Add proper legend, labels for axes, etc. Use `axis tight; after` the plot.
- In another plot, plot the number of Newton's iteration at each timestep, versus the index of the timestep. Add labels for axes. Use `axis([1 nstep 0.9 2.1]); after` the plot. See script `nlf1u.m` (in the course website) for some template.
- (d) [5 points] Do another simulation for time period from 0 to $t_{end} = 365$, with $h = 1/24$, and otherwise the same parameters as in (b), i.e. with $\rho = 300\mu$. Output the same quantities and do the same plots as in (c), with the same axis specifications.
- (e) [15 points] Do another simulation for time period from 0 to $t_{end} = 365$, with $h = 1/24$, and with the same parameters as in (b) (i.e. with $\rho = 300\mu$), except that $\beta = 0.25/4$, as if measures are taken to reduce contacts by a factor of 4. Output the same quantities and do the same plots as in (c), with the same axis specifications.
- In addition, in a another plot, plot $x_2^{(i)}, x_3^{(i)}$, versus $ih, i = 0, \dots, nstep$, in one plot, using dashed (green) and dashed-dotted (blue) lines. Add proper legend, labels for axes, etc. Use `axis tight; after` the plot.
- Place the first plot from (c) and the first from (d) side-by-side. Place the first and last plots from (e) side-by-side. Place the three plots of number of iterations side-by-side in 1×3 format. Comment on all results of (c), (d) and (e), including output and plots.**

Notes: Do not use any symbolic calculation of any sort. The derivatives should be derived by hand and hard-coded into the Jacobian matrix.

2. [To be done by hand/latex, no coding.] Let $f(x) = x \sin(x) - 1$.
- (a) [1 points] Determine graphically the number of roots of $f(x) = 0$ in the interval $[0, \pi]$. (Hint: Rewrite equation $f(x) = 0$ to make graphing by hand easy.)
- (b) [2 points] Show mathematically that f has a unique root in $[\frac{\pi}{4}, \frac{\pi}{2}]$.
- (c) [1 points] Consider the fixed-point iteration scheme $x^{(k+1)} = g(x^{(k)})$, with $g(x) = \frac{1}{\sin(x)}$. Show mathematically that g has a unique fixed point, say x^* , in $[\frac{\pi}{4}, \frac{\pi}{2}]$.
- (d) [8 points] Find a (closed) interval I of length at least $\frac{\pi}{6}$, that contains x^* , so that, if $x^{(0)} \in I$, the iteration $x^{(k+1)} = g(x^{(k)})$ converges to x^* . Specify I and explain. (You can specify the endpoints of I in terms of π .)
- (e) [4 points] What is the order of convergence of the iteration scheme in (d)? Justify mathematically.
- (f) [6 points] Show that the iteration $x^{(k+1)} = g(x^{(k)})$ converges to x^* , if $x^{(0)} \in [\frac{\pi}{4}, \frac{\pi}{2}]$. (Note: If you have already shown this in (d), you can skip this question.)
- (g) [3 points] Apply by hand one Newton iteration to $f(x) = 0$, with starting guess $x^{(0)} = \frac{\pi}{2}$, and indicate the computed $x^{(1)}$.

3. [To be done by hand/latex, except (e).]

- (a) [3 points] Use Newton's Divided Differences, to construct a interpolating polynomial $p_1(x)$ of the function $f(x) = e^{-x}$ (i.e. $f(x) = \exp(-x)$), based on the data points $x_0 = -1$ and $x_1 = 1$. Show the NDD table, and the interpolating polynomial $p_1(x)$.
- Note: It is fine to leave quantities such as e^{-1} and e in the presentation of the table entries and polynomial coefficients, as long as you reasonably simplify all terms.

- (b) [3 points] Give the polynomial interpolation error formula for this problem. Note that the formula involves an unknown point ξ .
Using the formula, find a (sharp) bound of the (absolute value of the) error for any $x \in [-1, 1]$. Indicate the interval ξ belongs to, when $x \in [-1, 1]$.
Using the formula, find a (sharp) bound of the (absolute value of the) error for any $x \in [1, 2]$. Indicate the interval ξ belongs to, when $x \in [1, 2]$.
- (c) [4 points] Use Newton's Divided Differences, to construct the lowest degree interpolating polynomial $p_2(x)$ of the function $f(x) = e^{-x}$ (i.e. $f(x) = \exp(-x)$), based on the data points $x_0 = -1$, $x_1 = 0$ and $x_2 = 1$. Show the NDD table, and the interpolating polynomial $p_2(x)$.
Note: It is fine to leave quantities such as e^{-1} and e in the presentation of the table entries and polynomial coefficients, as long as you reasonably simplify all terms.
- (d) [5 points] Give the polynomial interpolation error formula for this problem. Note that the formula involves an unknown point ξ .
Using the formula, find a (sharp) bound of the (absolute value of the) error for any $x \in [-1, 1]$. Indicate the interval ξ belongs to, when $x \in [-1, 1]$.
Using the formula, find a (sharp) bound of the (absolute value of the) error for any $x \in [1, 2]$. Indicate the interval ξ belongs to, when $x \in [1, 2]$.
- (e) [10 points] Give the Taylor polynomials $t_1(x)$ and $t_2(x)$ of degree 1 and 2, respectively, that arise by expansion of $f(x) = e^{-x}$ about $x = 0$. Write a MATLAB script that uses `polyfit` to compute the polynomials $p_1(x)$ and $p_2(x)$, interpolating $f(x)$ at 2 and 3 equidistant data points in $[-1, 1]$, respectively. The script then evaluates the four polynomials, p_1 , p_2 , t_1 , t_2 , and the function f at 100 equidistant evaluation points in $[-1, 1]$. Let v_1 , v_2 , u_1 , u_2 and v_f be the vectors of values of p_1 , p_2 , t_1 , t_2 and f , respectively, at 100 equidistant evaluation points in $[-1, 1]$. In one plot, graph v_1 , v_2 , t_1 and t_2 and v_f versus the evaluation points. Use solid, dashed, dotted, dotted-dashed and solid lines, respectively, and appropriate legend and labels. In another plot, graph $v_f - v_1$, $v_f - v_2$, $v_f - u_1$, $v_f - u_2$, versus the evaluation points. Use solid, dashed, dotted and dotted-dashed lines, respectively, and appropriate legend and labels. Calculate and output $\max_{\text{eval pts}} |v_f - v_1|$, $\max_{\text{eval pts}} |v_f - v_2|$, $\max_{\text{eval pts}} |v_f - u_1|$, and $\max_{\text{eval pts}} |v_f - u_2|$. Place the plots side-by-side with appropriate captions/subcaptions. **Comment on the results.**
4. [10 points] Consider using three types of interpolation for the function $f(x) = \exp(-x)$, namely, polynomial, linear spline and cubic spline interpolation, as in script <http://www.cs.toronto.edu/~ccc/Courses/336/scriptint.m>
Add the appropriate extra lines to the script to compute the cubic spline interpolant at the points indicated. Then run the script and collect the output. (If you get a warning from `polyfit`, you are not necessarily doing anything wrong.) **Comment on the results**, in particular, **how the error of each interpolant behaves as the number of data points increases, and whether this behaviour agrees with what expected from theory.**

Do not change the format according to which the errors and other quantities are output. You need to explain what the other quantities represent.

General note: Plotting quantity y versus quantity x , means that x is in the x -axis and y is on the y -axis, i.e. what follows "versus" is in the horizontal axis.