COMP5930M - Scientific Computing

Coursework 1

October 21, 2020

Deadline

10:00, Thursday 12th November

Task

The three numbered sections of this document describe problems which can be modelled by nonlinear equations. In each case a proper formulation of the problem as a nonlinear system is required with appropriate initial data. You will use one or more of the algorithms we have covered in the course to produce a numerical solution.

MATLAB scripts referred to in this document can be downloaded from Minerva under `Learning Resources / Coursework`. MATLAB implementations of some algorithms have been provided as part of the course but you can implement your solutions in any other language if you prefer.

You should submit your answers as a single PDF document via Minerva before the stated deadline. MATLAB code submitted as part of your answers should be included in the document. MATLAB functions should include appropriate `help` information that describe the purpose and use of that function.

Standard late penalties apply for work submitted after the deadline.

You should properly cite any sources of information used outside of the course material.

Disclaimer

This is intended as an individual piece of work and, while discussion of the work is encouraged, plagiarism in any form is strictly prohibited.

1. A floating sphere

[13 marks total]

A solid sphere of radius a with density ρ_s, assumed to be less than the density of water $\rho_w = 1$, will float in an infinite bath of water partially submerged at a depth H below the surface.

Archimedes' Principle:

> "*Any floating object displaces its own weight of fluid*",

allows us to compute the depth for a sphere of given radius and density.

The weight of water W_w displaced by the sphere is computed from the volume V submerged to depth H:

$$W_w \;=\; \rho_w V_{cap} \;=\; \pi H^2 \left(a - \frac{H}{3}\right).$$

The weight of the sphere W_s is

$$W_s \;=\; \frac{4}{3}\pi \rho_s a^3.$$

(a) Write in mathematical notation the nonlinear equation $F_{a,\rho_s}(H) = 0$ that can be used to solve for the submerged depth H for given values of a and ρ_s. Provide a MATLAB implementation of this function.

(b) Choose an appropriate initial bracket $[H_L, H_R]$ to be used in the bisection method that is guaranteed to contain the solution of the equation $F_{a,\rho_s}(H) = 0$. Show that this bracket is suitable for the bisection method by appealing to properties of F.

(c) Solve numerically the nonlinear equation using the bisection method and Newton's method for the case $a = 1$ and $\rho_s = 0.45$. Select a suitable initial guess x_0 for Newton's method that guarantees that both methods converge to the same solution. How many iterations are required when using $tol = 10^{-6}$ for both methods?

(d) How many different solutions can Newton's method converge to when you vary the initial guess x_0? Find different initial guesses x_0 such that Newton's method converges to each of these solutions.

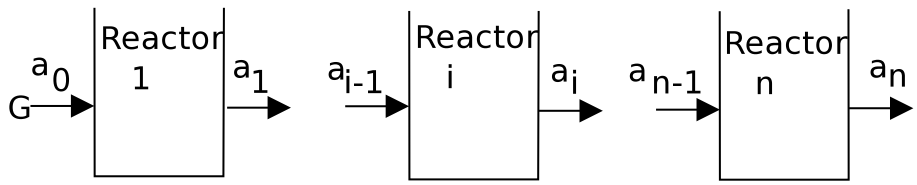

Figure 1: A series of reactor vessels

2. **Chemical engineering**

[14 marks total]

Figure 1 shows a series of n continuously-stirred reactor vessels, each of volume V, that are used for a particular chemical reaction. Materials are fed into reactor 1 at rate G with concentration a_0. Within the i'th reactor in the series, the chemical has the concentration a_i. The differential equation for the rate of change of the concentration $a_i(t)$ at time t is:

$$\frac{da_i}{dt} = -\beta a_i^2 + a_{i-1} - a_i, \qquad \text{for } t > 0$$

where $\beta = \frac{kV}{G}$. Here $-\beta a_i^2$ is a nonlinear term representing a chemical reaction that transforms the chemical to other compounds, and the term $a_{i-1} - a_i$ represents the change of concentration due to the flows in and out of the i'th reactor.

If we are only interested in the long-time response of the system, we can look for a *steady-state solution* by setting $\frac{da_i}{dt} = 0$ and obtain a system of n nonlinear equations for the steady-state concentrations a_i in the n reactors as

$$\beta a_i^2 - a_{i-1} + a_i \;=\; 0, \quad i = 1, 2, \ldots, n. \tag{1}$$

We assume that values for V, G, a_0 and k are known constants.

(a) For the case $n = 5$ reactors, write in mathematical notation the nonlinear equation system $\mathbf{F}(\mathbf{U}) = \mathbf{0}$ for (1). Implement the system in MATLAB code.

(b) Analytically derive the Jacobian matrix for (1) and provide the mathematical formulas for all the nonzero elements. Implement the Jacobian in MATLAB code.

(c) Produce a numerical solution for the $n = 5$ case with the following parameters: $V = 1.0$, $G = 35.0$, $k = 0.6$, $a_0 = 6.0$. using Newton's method with the exact Jacobian and $tol = 10^{-12}$. Report the values of $|\mathbf{F}|$ at each iteration.

(d) The unsteady problem is: find the time-dependent concentrations $a_i(t)$ s.t.

$$\frac{da_i}{dt} = -\beta a_i^2 + a_{i-1} - a_i, \qquad \text{for } t > 0$$

given an initial state $a_i(0) = a_i^0$ for for $i = 1, \ldots, n$.

Discretise this unsteady problem in time using the backward Euler (implicit) method and a time step of Δt. Write the nonlinear equations to be solved in each time step.

(e) Analytically derive the Jacobian matrix for the unsteady problem after discretisation with the backward Euler method and provide the mathematical formulas for all the nonzero elements. What happens to the Jacobian matrix when Δt approaches 0 ($\Delta t \to 0$)? What about when the Δt is made arbitrarily large ($\Delta t \to \infty$)?

3. **Control of a robot arm**

[13 marks total]

A simple robot arm consists of two rigid, straight segments of length 1, with the first fixed at the origin at one end and the two segments hinged such that the second can move freely. The location of the free end (loc_x, loc_y) is defined by the angles θ and ϕ that the arms make with the horizontal, such that the location can be defined as

$$\begin{aligned} loc_x &= \cos(\theta) + \cos(\phi) \\ loc_y &= \sin(\theta) + \sin(\phi) \end{aligned}$$

(a) Formally state the system of nonlinear equations to be solved, in the form $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, such that the angles $(x_1, x_2) = (\theta, \phi)$ can be computed to position the free end of the arm at a given location (loc_x, loc_y).

(b) Implement in code a function that represents this problem as a system of nonlinear equations in the form $\mathbf{F}(\mathbf{x}) = \mathbf{0}$.

(c) Using the Matlab function `newtonSys.m` show that, in general, two roots exist for this system by considering a target location $(loc_x, loc_y) = (0.5, 0.5)$ and initial states for your nonlinear solution algorithm:

   i. $(x_1, x_2) = (-1, 1)$

   ii. $(x_1, x_2) = (2, 0)$

   State the solutions obtained in each case. Use the Matlab function `showArm.m` to display the computed configuration in each case and include the figures produced in your report.

(d) The free end of the arm is required to trace the path defined by the following function

$$(x, y) = (t, 0.1 + \sin(2t - 0.5)), \qquad t \in [-1, 1]$$

   using 21 equally spaced steps across the range of t.

   Write a Matlab function, named `traceArm.m` with the following definition

   `traceArm( traceFn, nSteps, x0 )`

   where `traceFn` is the function defining the path, `nSteps` is the number of steps required to go from $t = -1$ to $t = 1$, and `x0` is the initial guess for the free end.

   i. Explain how you select the initial state x_0 at each step of the algorithm.

   ii. Create a table showing the angles θ and ϕ at each step.

Learning objectives

- Formulation of nonlinear equation systems.
- Implementation of nonlinear systems in MATLAB.
- Application of standard solution algorithms for nonlinear systems.
- Definition of initial solutions based on problem-specific knowledge.
- Convergence properties of Newton's method.

Marking scheme

There are 40 marks in total. This piece of work is worth 20% of the final module grade.

1. A floating sphere [13 marks total]

   (a) [2 marks] nonlinear function and MATLAB implementation
   (b) [3 marks] the initial bracket plus justification for its validity
   (c) [4 marks] two correct solutions plus the # of iterations
   (d) [4 marks] number of solutions and corresponding initial guesses

2. Chemical engineering [14 marks total]

   (a) [3 marks] nonlinear system for the steady-state problem and MATLAB code
   (b) [3 marks] Jacobian matrix for the steady-state problem and MATLAB code
   (c) [2 marks] numerical solution and values of $|\mathbf{F}|$
   (d) [2 marks] discretised time-dependent equations
   (e) [4 marks] Jacobian matrix for the time-dependent problem, behaviour for limit values of the time step

3. Control of a robot arm [13 marks total]

   (a) [2 marks] system of equations
   (b) [2 marks] MATLAB implementation
   (c) [3 marks] two correct solutions, figure
   (d) [6 marks] implementation of the path and the tracing function, the table of angles