Modelling and Programming — IT & Health

Week 4

Report 1 Deliverables

- Report 1: Assignments 13–15.

The MATLAB quiver plot

To solve Assignment 13, you need to draw a quiver plot using MATLAB. Drawing a quiver plot means to draw the little arrows that you see in Figures 11.2 and 11.4–11.8 of **C**. This is a slightly advanced procedure, but you can get started by writing a MATLAB program that implements the following algorithm:

1. Construct five variables to describe the begin and end values of the two axes in the coordinate system and the number (`n`) of arrows along each axis. Call them `x_begin`, `x_end`, `y_begin`, `y_end`, and `n`. Set `n` to `30` and, as a first example, use `-5` and `5` as begin and end values.
2. In cases where the x and y axes differ in length, the arrows will not be properly sized. To handle such cases, construct a variable `y_scale` which is equal to the length of the x-axis divided by the length of the y-axis.
3. Construct a vector called `x` that contains `n` values between `x_begin` and `x_end` and a vector called `y` that contains `n` values between `y_begin` and `y_end`.
4. Use the MATLAB command `meshgrid` to construct variables `X` and `Y` that contain x and y coordinates for every arrow to be drawn in the coordinate system. (*Hint*: Write `doc meshgrid` in the MATLAB command window for help.)
5. Construct a matrix of quiver vectors called `qvec` which is filled with 1's. The matrix must hold one 2D vector for each arrow, that is, it should be of dimensions $n \times n \times 2$. (*Hint*: See Section 5.4 of **P**.)
6. Extract the u and v vector components from `qvec` and store them in vectors called `u` and `v`, respectively. (*Hint*: Do indexing using the : operator.)
7. Scale the vectors `y` and `v` by `y_scale`.
8. Use the MATLAB command `quiver` to draw the arrows defined by the quiver vectors in `qvec`. (*Hint*: Write `doc quiver` in the command window for help.)
9. Beautify your plot using the following commands:
```
xlabel('x_1');
ystring = sprintf('scaled x_2 = %.1f \times x_2', y_scale);
ylabel(ystring);
axis([x_begin x_end y_begin*y_scale y_end*y_scale]);
```

Make sure that you understand why all the arrows point up and to the right.

Direction fields

Assignments 13 and 14 are about drawing and using a direction field to explain the dynamics of a process which is described by a system of differential equations. The MATLAB quiver plot is used to draw direction fields, but you need to insert a step in the algorithm which sets the vectors in `qvec` to the *direction vectors* given by the differential equations. This step should be inserted between Steps 5 and 6 of the algorithm. Do the following exercises to get started on drawing direction fields:

- Construct a 2×2 matrix `A` that represents the matrix form of two differential equations (use Equations 11.5 in Section 11.1.1 of **C** as a first example).
- Insert a double `for`-loop between Steps 5 and 6 of your quiver plot implementation which computes direction vectors for `qvec` using the matrix `A` and indexing into the matrices `X` and `Y`. (*Hint*: Take a look at Section C.5 of **P** to brush up on `for`-loops and indexing.)
- **C**: Problem 9 in Section 11.1.4. Use your quiver plot implementation. Ensure that your program reproduces Figures 11.18–11.21 correctly. In this way, you have tested your implementation.

Now, solve Assignments 13 and 14.

Eigenvalues and eigenvectors

Assignment 15 is about using eigenvalues to determine the development of a dynamical system in the long run. To get a better understanding of eigenvalues and eigenvectors, do the following exercises:

- **C**: Problems 75–76 in Section 9.3.4. Use the MATLAB command `eig` to find the eigenvalues and eigenvectors. (*Hint*: Closely related to **C**: Pages 558–559.)
- **C**: Problems 57–58 in Section 11.1.4. Use the MATLAB command `eig` to find the eigenvalues and eigenvectors and reproduce Figures 11.31 and 11.32 using your quiver plot implementation.

It is important to notice what eigenvalues and eigenvectors tell you about the long-term behaviour of a dynamical system.

The differential equations in the report assignment are not linear. This means that the long-term behaviour of the system cannot be determined from a single computation of eigenvalues and eigenvectors. The reason is that the matrix (**A**), which describes the system, changes for each time step. At the outset, or after a disturbance, a nonlinear dynamical system experiences a *transient* behaviour. However, the transients often die out over time. By computing eigenvalues for the matrix that describes a nonlinear system after each time step, we can determine whether the eigenvalues become constant over time. Once the eigenvalues approach constant values, we can stop the simulation and predict the long-term behaviour of the system by analysing the constant values that the eigenvalues are approaching.

Now, solve Assignment 15.

Curriculum

C	Sections 11.1.1–11.1.3 without Case 2. *Direction Fields* and *Equilibria and Stability*.
LA	Sections 3–3.2.1. *Eigenvalues and Eigenvectors*.
P	Appendix C. *The Operators* $*$, *^, / and \ for Matrices and Vectors*.

JRF 2009