CA205 Assignment 1: Review Notes

Greg Gamble*

email: gregg@maths.uwa.edu.au
WWW: http://maths.uwa.edu.au/~gregg/CA205/

October 3, 1997

1 Writing Skills

The content of submissions was lower than I expected, with the main weaknesses being in selection and ordering of material. In Section 5, a *benchmark structure* for assignments is given, in the hope that students will have a better idea of how to present their *next* assignment! The 500 word report should be readable in its own right, like a front page newspaper article, with pointers to supporting material; the "what happened" and "why is it so" rather than the "how did I do it". Program source code and calculations fall into the latter, graphs and tables generally also, though there may be compelling reasons to include particularly informative graphs in the main report body.

Referencing and citation skills were poor. Reference lists are used by the reader to assess the technical reliability of the material. Books require names, title, publisher, and year at the very minimum; look at some bibliographies for examples. All documents need to be referred to by title, web pages by the URL (`http://...`). Students need to have their student number cited, and staff have their affiliation presented.

Write as if this report was being read by a senior manager, who doesn't have a lot of time, and is depending upon your presentation to make a decision.

2 Technical Aspects

The connection to physical reality was generally poor. As indicated on the first assignment sheet, the constants K and h have physical interpretations, the nature of which motivates a sensitivity analysis. The somewhat arbitrary nature of calculation is also another indicator driving a sensitivity analysis.

Whilst the iron was modelled as being in two states, it wasn't clear how these states related to the actual iron – was a piece of metal in a different place? Was there a change in polarity? Without something physical to relate to, the thermostat programming was quite arbitrary.

Error analysis – the "correct answer" needs to be defined.

*Much of this was pirated from a similar document by Patrick C Hew (phew@maths.uwa.edu.au)

3 Other

Please place tutor names on the covers, as this was the primary key for sorting. It is to your benefit if a tutor can put a face to the writeup.

4 Perceptions

Students were requested to submit a suggested marking key. Comparison with the actual key used indicated good goal setting skills.

5 Assignment Structure

When we were highschool students we were taught in Chemistry and Physics a basic structure for writing up experiments – something along the lines of Aim, Procedure, Discussion. Something like this basic structure should have been followed for Assignment 1. If one looks up journals in the library one finds this structure is taken a bit further, and this is the structure you as students should strive to attain. A typical journal article structure is as follows (the description of the components has been modified to be as relevant as possible to Computer Applications students).

Abstract: A heading **Abstract** should be followed by two or three sentences that *very briefly* describe the *Aim*, *Procedure* and *Conclusion*. There should be *no* specific references to the text of the report or elsewhere. It should be written *last*, despite the fact that it appears *first*. For example, an *Abstract* for Assignment 1 might have said:

> A computer model of the sole-plate temperature of an electric iron is developed. First the iron sole-plate temperature is modelled mathematically by a linear first order linear differential equation and approximated numerically via the Euler method. It was found that a reasonable model could be achieved by incorporating a modelling of the iron's thermostat.

Introduction: This should have a heading **Introduction**. Much of what is stated in *Abstract* is repeated here. Again the *Aim* and the *Procedure* are given, but this time with specific references to the *Body* and *Appendices* of the text. This section serves to link the report together, and should be the *second-last* component of the report to be written.

Body: The *Body* should be broken up into sections, each with their own descriptive heading, e.g. Assignment 1 might have the following sections:

> **Determination of the constants h and K.**
> **Incorporation of thermostat in model.**

Conclusion: This should have a heading **Conclusion** followed by a discussion of results. This is also where you should include your paragraph of *what you think you have learned.*

Acknowledgements: If there are *general* acknowledgements these should appear in a section headed **Acknowledgements**. There can also be acknowledgements spread through other sections of the report.

Bibliography: Most reports should have a bibliography with heading either **Bibliography** or **References**, followed by a list of references. There are many styles of presentation but the following example exhibits one way to list references:

[1] D. E. Taylor, *The geometry of the classical groups*, in *Sigma Series in Mathematics* **9** (Heldermann, Berlin, 1992).

[2] H. Wielandt, *Finite permutation groups* (Academic Press, New York, 1964).

Most references a Computer Applications student might wish to include in a bibliography will be books and will follow the pattern of [2].

Appendices: Each *Appendix* should have a descriptive heading, e.g. one appendix that could have appeared in Assignment 1 is

Appendix A: Experimental Heating and Cooling Data.

6 Some comments on Boolean variables

In modelling the thermostat many students had rather convoluted code. Also, many students used `state` or some other undescriptive variable name to represent the state of the iron. Since the iron could only be in one of two states: `on` or `off`, a better name for such a variable would be something like `on` or `heating`. Aside from this, a number of students submitted code along the lines of:

```
if (state == 1)
      ⋮
     state = 0;
      ⋮
end;
if (state == 0)
      ⋮
     state = 1;
      ⋮
end;
```

Now since the two `if`-blocks are executed sequentially there is a potential problem that the second `if`-block will simply undo any change made by the first `if`-block. As it turned out, all students who had such code had sufficient code in the omitted code (represented by ⋮) to prevent this. Nevertheless, the middle two lines of the above code should *most definitely* be encoded as:

```
else
```

It might also be of interest to know about the following technique, which we explain by means of an example based on Assignment 1. We have variables `t` (time in seconds), `T` (temperature of the iron sole-plate in °C) and `on` (Boolean variable representing the on/off state of the iron heating element). We also have the following additional information: the iron is turned on at $\mathtt{t} = 0$ and turned off at $\mathtt{t} = 900$, the thermostat has cut-out temperature 370°C and cut-in temperature 250°C. Thus

> $\mathtt{t} < 900$ or $\mathtt{t} \geq 900$; and
> $\mathtt{T} \leq 250$ or $250 < \mathtt{T} < 370$ or $\mathtt{T} \geq 370$; and
> $\mathtt{on}_{\text{prev}} = \mathtt{true}$ or $\mathtt{on}_{\text{prev}} = \mathtt{false}$.

Overall there are $2.3.2 = 12$ possible intersections of these conditions. We are interested in discovering how to represent $\mathtt{on}_{\text{next}}$ as a Boolean function of $\mathtt{t}, \mathtt{T}$ and $\mathtt{on}_{\text{prev}}$. It is convenient to represent the universal set of possibilities as a rectangle as follows (the idea is similar to a Venn diagram you learnt about in school – this representation is called a Carroll diagram, and is essentially what Electrical Engineers know as a Karnaugh map). For convenience we abbreviate $\mathtt{on}_{\text{prev}} = \mathtt{true}$ to $\mathtt{on}_{\text{prev}}$ and $\mathtt{on}_{\text{prev}} = \mathtt{false}$ to $\mathtt{off}_{\text{prev}}$, etc.

	$\mathtt{t} < 900$		$\mathtt{t} \geq 900$
$\mathtt{T} \leq 250$	$\mathtt{on}_{\text{next}}$	$\mathtt{on}_{\text{next}}$	
$250 < \mathtt{T} < 370$	$\mathtt{on}_{\text{next}}$	$\mathtt{off}_{\text{next}}$	$\mathtt{off}_{\text{next}}$
$\mathtt{T} \geq 370$	$\mathtt{off}_{\text{next}}$	$\mathtt{off}_{\text{next}}$	
	$\mathtt{on}_{\text{prev}}$	$\mathtt{off}_{\text{prev}}$	

Observe that there are only 7 regions represented above (not 12 as indicated there should be, above); the region $\mathtt{t} \geq 900$ should be divided into 6, according to our discussion – but since each such subdivision would have $\mathtt{off}_{\text{next}}$ in it we have rationalised it to 1 region. From the above diagram we obtain in Matlab parlance

```
on = (t < 900) & ((T <= 250) | (on & (T < 370)));
```

Note that this is exactly equivalent to the encoding

```
if ((t < 900) & ((T <= 250) | (on & (T < 370))))
     on = 1;
else
     on = 0;
end;
```

... but the former is both more efficient and more *elegant*!