

Coding standards and requirements

This is a list of computing and coding specifications. Please adhere to these for coding assignments. Careful coding may seem to be a waste of time, but experience shows that it saves massive time in the long run. Finding bugs and tiptoeing around fragile code take much longer than following these coding guidelines. You may lose points on programming exercises if you do not follow these guidelines.

1. Do not *hard wire* parameters or constants. If you have a formula $u = (i/m)p$ and $p = .3$, do not implement it as `u = (i/m)*.3`. Instead, define a variable and set it to `.3`, as ...

```
p = .3          # red ball probability
u = (i/m)*p     # up probability
```

This makes the code easier to read. It protects against certain mistakes, particularly if p occurs more than once, as in $v = ((m - i)/m)(1 - p)$. It allows you to change the value of p more easily.

2. Comment any piece of code you add. It does not have to be as thick as the prototype codes posted, but any functionality you add or change should be commented at least a little. This applies to something as small as putting another curve in a plot figure. Comment on the role of any variable you add when it is defined if the name does not make its role obvious, and possibly even if it does (see point 1. above).
3. When you modify a plot, also modify the legend and title to keep everything consistent. If you go from Brownian motion to Ornstein Uhlenbeck, you have to change the plot title. If you put curves from multiple times, the times must be indicated either in the title, the subtitle, or legend.
4. When you modify existing code, follow the coding style of the code you are modifying. Use “white space ” (blank spaces and blank lines) in the same way the original code does. It should not be possible to look at the code and see from the coding style which lines are new.
5. When you modify a plot, make sure the new version is at least as automated as the old version. Do not add labels or legends to plots by hand. Have the code do it.
6. You should hand in graphs or output numbers to illustrate whatever remarks you make. If you say that curves are converging, or agree, there should be plots of enough of these curves to see that.