Criteria	0-39% (F, D-) Poor	40-49% (D, D+) Satisfactory	50-59% (C-, C, C+) Good	60-69% (B-, B, B+) Very Good	70-89% (A-, A, A+) Excellent	90-100% (A*) Outstanding
Completeness	Significant parts of the requested method are missing. The code provided has only a fraction of the functionality required.	The bare minimum to provide core functionality has been implemented. Significant issues with Input/Output (I/O).	Methodology provided has been mostly implemented, providing little more than the core functionality. I/O may be insufficient, erroneous or superfluous.	Methodology provided has been implemented fully. Some ancillary features are missing. I/O may be insufficient, erroneous or superfluous.	Methodology provided has been implemented fully. All requested features are present. All inputs & outputs available to the user as requested/in the most appropriate form.	Very little could be done to bettered it in any way
Readability	Excessive time required to comprehend what the code is meant to do. Code is for all intents and purposes unreadable.	Excessive time required to comprehend the code supplied but the structure and some comments make comprehension possible.	The majority of the code is comprehensible with moderate effort. Comments and help entry sufficient to partially assist in comprehending what the code does. Some issues with the variable and function identifiers used.	Spacing and indentation used throughout most of the code. Appropriate function and variable identifiers mostly used. Appropriate level of commenting present. Complete help entry provided. Very little effort is required to understand what the code does.	Spacing and indentation used throughout the code and is consistent. Appropriate function and variable identifiers used throughout. Appropriate level of commenting present. Complete help entry provided. We can understand what the code does at a glance.	
Elegance	No intuitive approach used for the solution of the problem. Solution is excessively complex for no reason. Significant amount of unnecessary code present.	Very little consideration was given to the efficiency of the code. The solution presented is overly complex but the reason for choices made is evident. Some unnecessary code is present.	The code is overall efficient but in places excessive memory usage or runtime were identified. A complex approach was utilised but while not the most elegant of solutions is reasonable in the context of making a rough prototype implementation.	Code is efficient, not requiring excessive memory or runtime usage but could be further improved. Not overly complex in implementation. More appropriate methods could have been used in places for a clean, robust and concise solution.	Code is very efficient, not requiring excessive memory or runtime usage. Not overly complex in implementation. Most appropriate mixture of methods for a clean, robust and concise solution utilised.	
Consistency	Code shows very little understanding or awareness of the datatypes used and/or how arrays/scalars work. Significant inconsistencies evident in the implementation of the method, units used etc.	Variable datatypes not maintained in certain cases and methods chosen in several places inappropriate to the datatype used. Scalars and arrays often not treated appropriately. Several dimensional or unit inconsistencies observed.	Variable datatypes mostly maintained and methods chosen in several mostly appropriate to the datatype used. Scalars and arrays mostly treated appropriately. Few or no dimensional or unit inconsistencies observed.	Variable datatypes maintained throughout but methods chosen in places inappropriate to the datatype used. Scalars and arrays treated appropriately throughout. No dimensional or unit inconsistencies observed.	Variable datatypes maintained throughout and methods chosen are appropriate to the datatype used. Scalars and arrays treated appropriately throughout. No dimensional or unit inconsistencies observed.	
Correctness	Code fails to correctly implement the method provided. The code either fails to run (syntax or runtime issues) or consistently returns an incorrect answer.	No syntax errors but significant logical or runtime errors encountered in even the simplest of test cases.	No syntax errors but some minor logical or runtime errors encountered in some test cases.	No syntax errors but some minor logical errors encountered in most extreme of test cases.	No syntax errors. No runtime errors for all possible test cases. All outputs correct (no logical errors) for all possible test cases.	