Guidelines for 509 Questions (Introduction to Computational Science) for the CPSI MS Comprehensive Exam

1. Be able to answer questions about the limits of floating point representations in computers. Calculate rounding and representation errors given basic formulas for resulting sigma errors for addition and multiplication.

2. Create difference equation or recurrence relation representation of a described sequence. Be able to create pseudo code to generate the Nth value of a provided recurrence relationship. Answer questions to transform or calculate values of a recurrence relationship by hand.

3. Be able to answer questions about root finding methods and algorithms. For example, given recurrence relationship for the secant method, calculate next 3 terms for a given function, and calculate the percentage errors of these terms.

4. Use the Taylor Series to expand a function. Be able to compute an approximation of an expanded function to second, third, fourth order term. Be able to estimate approximation error given the function approximation expansion.

5. Explain and understand the basic methods for numerical integration, including Trapezoidal rule and Simpson's Rule methods. Demonstrate use of estimation of errors on integrals (based on Taylor expansions), and be able to apply this to implement adaptive integration versions of Trapezoid and Simpson's methods.

6. Be able to understand and apply the basic concept of finite differences to compute numerical derivatives, using backward and forward difference methods. Also be able to explain and demonstrate the central difference method and show and understand how this second order approximation gives better estimates than a first order forward or backward approximation.

7. Be able to apply Euler's method to solve ODE. Understand the relationship between Euler's method and Runge - Kutta solvers, and the increase in precision from using a second, third, fourth order Runge - Kutta approximation in the solution of an ODE.

8. Be able to answer questions about solutions of simultaneous linear equations using Gaussian elimination and back-substitution.

9. Demonstrate the basic creation of pseudo-random sequences using a Mersenne twister. Be able to create pseudo code implementations of simple Monte Carlo simulations to solve probability problems, like expected number of balls drawn from a bag of balls with or without replacement, throwing dice, card games or other events.