1 ALGORITHMS

Primary

1. Monte Carlo Integration
2. Taylor approximation
3. Forward and centered difference approximations to derivative
4. Interpolation - with nested multiplication
    - Power form (Vandermonde matrix)
    - Newton form
5. Matrix construction by diagonals and blocks
6. Matrix operations
7. 2-D function tables, contour plots

Secondary

1. Stirling's approximation to $n!$
2. Decomposition of wave into sines (or cosines)
3. Construction of special matrices (Pascal, Toeplitz, etc.)
4. 2-D quadrature

Know what each algorithm is used for and how it works; be prepared to carry out the steps of algorithms (with perhaps the help of a calculator) in simple cases.

2 ANALYSIS

1. Accuracy - absolute and relative error
2. Efficiency - big-O notation, flop count
3. IEEE long precision binary floating point number system
   - $v(s, e, f) = (-)^s * (1.f)_B * 2^{e-1023}$
   - `eps, realmax, realmin`
4. Effect of roundoff

Know what the analysis (presented in the text or demonstrated in class) says about each of the algorithms above. Be prepared to apply that analysis in specific cases.

3 SOFTWARE

1. Memory management
2. Vector and matrix operations - Vectorization!
3. Functions, Plotting
4. Program control: `for, if-else, while`
5. Benchmarking

Be familiar with these computing concepts as they are realized in MATLAB. Be prepared to deduce what a given fragment of MATLAB code does, and to modify or extend a given code fragment to improve its efficiency or produce a desired different result.