9 Summary of MA402

We now know how to solve some real problems using computers.

9.1 Mathematical Modeling

1. Modeling process (problem, physical laws, identify dependent/independent variables ...)
2. Newton's cooling (The cooling process of a couple of coffee ...)
3. Heat diffusion in a wire (1D), in Fin (1D and 2D), with little insulation (1D and 2D), with decay and advection (1D and 2D).
4. Flow of a pollutant in a stream (1D), in a lake/ocean (2D) with/without decay.
5. Mass transfer in two directions.
6. Fluid flow in 2D porous medium.
7. Ideal fluid flow

9.2 Numerical Analysis/Methods

1. Taylor expansions, the mean value theorem and and the extended mean value theorem.
2. Matrix, vector theory and operations ($A + B, AB, A^{-1}, Ax \cdots$).
3. Matrix and vector norms ($\|x\|_p, \|A\|_p$, $p = 1, 2, \infty$).
4. Gaussian elimination, backward/forward substitution, tridiagonal solver (trid.m).
5. Computer number system and operations ($fl(x) = x(1 + \epsilon)$), machine epsilon.
6. Round-off error analysis, absolute/relative errors. How do design certain algorithms to reduce the round-off error.
7. Introduction to ODE/PDEs
   - ODE/IVP, two-point boundary value problems, elliptic, parabolic, hyperbolic ODE/PDEs, boundary/initial value problems, classify second order partial differential equations.
   - Steady state solutions.
   - Dirichlet, Neumann, Robin/mixed boundary conditions and initial conditions.
8. Finite difference methods

- Basic procedure (grid, derivative $\Longrightarrow$ finite difference $\Longrightarrow$ system of equations).
- Forward, backward, central difference schemes for u_x.
- Central difference schemes for u_{xx}.
- Consistency and local truncation errors, stability, convergence of finite difference schemes, order of methods.
- Explicit and implicit time marching method

9. Finite difference methods for three types linear ODE/PDEs.
10. Basic stationary iterative methods: Jacobi, Gauss-Seidel, $SOR(\omega)$ methods and applied to ODE/PDEs

9.3 Programming, software, supercomputing and parallel computing

1. Matlab usage (super-calculator, visualization tool, computer language)
2. Programming basics (loops, sum ($\sum$), product $n!$) etc. How to make algorithms more efficient (i-j or j-i in matrix-vector multiplication.
3. Use of super-computers at North Carolina Super-computer center (Cray T916, IBM-SP2 SGI Orgin 2000).
4. Parallel computing: vector-pipeline shared memory and Open MP, Distributed machine and MPI

9.4 Reading and writing progress

- How to write a technical report and a paper.
- How to participate in a team work.