# Numerical Methods I

Review for Final Exam

Friday, December 12, 2003
9:30-11:30, Research II Lecture Hall

1. All the topics up to the Midterm Exam.
2. Polynomial interpolation: Know the Lagrange basis polynomials and be able to apply them (for example for deriving Newton–Cotes quadrature formulae, or Adams solvers for ODEs; problems like Questions 6.3 and 6.4 from the homework); understand more generally how interpolation with basis functions works (e.g. for Hermite or trigonometric interpolation)—you don't need to remember formulae, but should be able to use the ideas.
3. Spline interpolation: Basic ideas; be able to derive the system of equations that determines a spline (with some hints given); what type of system of linear equations do you get? Can you derive some of the minimal properties of splines?
4. Orthogonal polynomials: basic concepts.
5. Numerical Quadrature: Newton–Cotes formulae, Gauss quadrature; understand the idea and use of the Euler–Maclaurin summation formula (you do not need to remember its exact form); Romberg integration. Questions 8.1 and 8.2 could be similar to exam questions. (Do you realize that you have derived a Gauss quadrature formula in Question 8.1?)
6. Basics of differential equations: Be able to explicitly solve simple equations (e.g. the logistic equation, linear $2 \times 2$ systems by diagonalization). What is a critical point, how to check its stability (Question 10.3)?
7. Matrix exponential: idea, compute the matrix exponential for simple matrices, basic application to ODEs.
8. Methods for solving ODEs:
   - Basic concepts: implicit vs. explicit, one-step vs. multistep.
   - Local truncation error (computing the local truncation error for some method *will* be on the exam!)
   - Convergence: consistent one-step methods always converge (why?), multi-step methods converge if and only if they are 0-stable.
   - Order of convergence
   - Absolute stability
   - Families of methods: Taylor, Runge–Kutta, Adams–Bashforth, Adams–Moulton (ideas only, no need to remember formulae)
   - Think about root finding schemes for implicit methods. When does simple iteration work?