MATH 5620 NUMERICAL ANALYSIS II
MIDTERM REVIEW SHEET

Chapter 7 – Iterative methods for linear systems

- Neumann series, sufficient condition for convergence, spectral radius (together with necessary and sufficient condition for convergence), induced matrix norm. Iterative refinement.
- Jacobi, Gauss-Seidel, SOR. Advantages, disadvantages and pseudocode. Relaxation methods and Chebyshev acceleration are not included.
- Steepest descent algorithm: you should know how to derive this (without memorizing it). Given a quadratic form, you should know how to find its minimum along a direction. SD uses the residual (negative gradient) as the direction.
- Conjugate gradient: do not memorize algorithm. You should be able to explain what are the advantages and disadvantages of this method with respect to direct methods. Why is CG a good method for sparse matrices? What is a preconditioner?

Chapter 5 – Initial Value Problems

- §5.1 Lipschitz condition, well-posedness, existence and uniqueness theorem for first order IVP.
- §5.2-5.3: Taylor series method, know how to derive them. Notion of order, local truncation error and global truncation error (in general). How to find local truncation error for Taylor series method.
- §5.4: Runge-Kutta methods. Advantages over Taylor series methods. Know general form of second order Runge-Kutta method, and how the parameters are related (match Taylor series). You don't need to know all the RK2 methods by their names. Why are RK methods of order higher than 4 less attractive than RK4?
- §5.5: Runge-Kutta-Fehlberg method. You have to know the basic idea (using embedded methods). Please do not learn the method by heart or its coefficients. Need to understand the derivation of update for h.
- §5.6 Multistep methods. Understand the relationship with integration formulas. Use the undetermined coefficient method. Difference between Adams-Bashforth and Adams-Moulton. What is a predictor-corrector method? How is it related to a fixed point iteration?

- Analysis of linear multi-step methods. Know the linear functional L and how it can be used to get the order or the local truncation error of a method. You do not need to know the formulas for the d_j exactly, but you should know where they come from.
- §5.7 Variable step-size multistep methods. You need to know the big idea to obtain adaptivity. How does this method compare to Runge-Kutta-Fehlberg from the practical point of view? As for RKF you may be asked to estimate the size h to get a desired precision.
- §5.8 Extrapolation methods: Know the main idea (clever linear combinations to get error cancellations) and how to apply it to an example.
- §5.9 Higher order equations and systems. Know how to transform an algorithm for the scalar case to system case. Know how to transform a higher order equation into a system.
- §5.10 Stability, only for *linear* multistep methods. Difference between implicit/explicit. Notion of convergent method. Consistency and stability notions and how they relate to convergence. Know how to determine if a given linear multistep method is consistent, stable or convergent. (you don't need to know the proof that convergence implies stability and consistency).
- Difference equations. Know how to solve a difference equation when roots of characteristic polynomial.

1. Other considerations

- Understanding of the methods will be favored over simple memorization.
- You should be able to evaluate a method based on its practical merits (does it take too many function evaluations?...)
- Expect to write pseudo-code
- You are expected to know how to solve 2×2 systems (and simple 3×3 or triangular systems). Also you should know how to find the roots of a quadratic or a simple cubic or quartic polynomial.
- A calculator is not needed for the exam, and is not allowed.
- A one-sided, letter sized, handwritten cheat sheet is allowed.
- No books or other notes are allowed.