Review Guide for In-Class Midterm Exam on Friday, 24 March 2023

The Midterm Exam on Friday, 24 March is *closed book* and *closed notes*. Please bring nothing but a writing implement.

It will cover Chapters 1, 2, 3, and 5 of the textbook,[1], plus sections 4.1 and 4.2 and some content from Appendices (see below), plus the "Classical iterative methods for linear and nonlinear systems" notes. On this Review Guide I list *specific* material that will be covered. Material significantly different from what is listed below will *not* be covered. My goal is to only include topics that have appeared on homework and in lecture.

The problems will be of these types: state definitions, state/derive formulas, explain/justify theorems and formulas, compute simple examples, give examples with certain properties, or describe/illustrate/sketch concepts. You are expected to use reasonable or common notation, or at least make your notation clear. Note that I will *not* ask you to "state definition 2.1" or anything like that which requires remembering locations in the book.

Strongly recommended: Get together with other students and work through this Review Guide. Be honest with yourself about what you do and don't know. Talk it through and learn! Also, please ask questions about Exam content during lectures on Monday 20 March and Wednesday 22 March.

Definitions and Notation. Be able to state and use the definition, and/or use the notation/language correctly:

- acronyms: ODE, PDE, IVP, BVP
- order of a differential equation (= maximum number of derivatives in it)
- one-sided and centered finite difference approximations of derivatives $u'(x)$ and $u''(x)$ (pages 3–4)
- absolute and relative error (Appendix A.1)
- $O(h^p)$ and other "big-oh" notation (Appendix A.2)
- vector norm (Appendix A.3)
- errors in grid functions (Appendix A.4)
- local truncation error (sections 2.5, 3.4, and 5.4: τ^h is the residual from applying the scheme to the exact solution)
- global or numerical error (section 2.6: $E_j = U_j - u(x_j)$ or $E = U - \hat{U}$)
- stable method (definition 2.1 in section 2.7)
- consistent method (section 2.8)
- convergent method (section 2.9)
- Poisson equation, Laplace equation, Laplacian operator (page 60, section 3.1)
- 5-point stencil for Laplacian in 2D (section 3.2)
- eigenvalues and eigenvectors (see Assignment #5: $Av = \lambda v$ where $v \neq 0$)

[1] R. LeVeque, *Finite Difference Methods ...*, SIAM Press 2007

- spectral radius (Appendix C)
- diagonalizable matrix (Appendix C.2)
- matrix exponential (Appendix D.3)
- Richardson, Jacobi, and Gauss-Seidel iterations for a linear system $Ax = b$ (Assignment #3 and "Classical iterative methods ..." slides, *and* sections 4.1, 4.2)
- Newton iteration (Assignment #3 and "Classical iterative methods ..." slides, *and* section 2.16)
- $f(u, t)$ is Lipschitz continuous in u (formula (5.15) in section 5.2)
- forward Euler, backward Euler, and trapezoid methods (section 5.3)
- explicit midpoint and explicit trapezoid methods (section 5.6; EM is (5.30)), as examples of Runge-Kutta methods

Formulas, Theorems, and Lemmas. Understand and remember.

- Taylor series (page 5) and Taylor's theorem with remainder
- fundamental theorem of finite difference methods (statement (2.22), section 2.9), which is essentially the Lax equivalence theorem
- convergence lemma for iterations $y_{k+1} = My_k + c$ (Assignment #3 and "Classical iterative ..." slides)
- definitions of the matrix exponential (Appendix D.3)

Techniques. Understand and remember. Be able to illustrate with an example. Be able to take a single step, or do a single iteration, on a simple example.

- derive a finite difference approximation by the method of undetermined coefficients (page 7)
- set up a finite difference scheme for a general, possibly nonlinear, second-order ODE BVP (sections 2.4, 2.15, 2.16.1)
- Newton's method (section 2.16.1, "Classical iterative methods ..." slides)
- set up a finite difference scheme for Poisson equation in 2D (sections 3.2, 3.3)
- implement a Neumann boundary condition for an ODE BVP FD scheme (section 2.12)
- convert a higher-order scalar ODE into a system of first-order ODEs (example given in section 5.1)
- take a step of the forward Euler, backward Euler, trapezoid, explicit midpoint, or explicit trapezoid method (sections 5.3 and 5.6; EM is (5.30))
- draw an explanatory sketch or cartoon of a Runge-Kutta method, from its formulas (as done in lecture)

Make sure you can do these techniques! Practice explaining/showing examples to another person. During the Exam the emphasis will be on quickly stating definitions, or setting-up or explaining techniques, or doing brief computations, all on paper and considering only easy examples. Pseudocodes will be requested in some cases, but you will *not*, of course, be asked to do many steps of anything.