Exam topics "MATH-251(a) Numerical Analysis", Fall 2024

1. Nonlinear Equations

   - Understand and implement bisection method.
   - Implement fixed point methods and analyze their local convergence (convergence rate, convergence order)
   - Apply and implement Newton method for finding solution of a nonlinear equation or a system of nonlinear equations

2. Curve Fitting

   - Perform polynomial interpolation of given data in monomial basis as well as basis of Lagrange polynomials
   - Implement polynomial interpolation, including the evaluation of the interpolating polynomial
   - Understand the implications of Theorem 2.2 for the behavior of the interpolation error
   - Understand the advantages of Clenshaw-Curtis nodes for convergence and stability of interpolation
   - Understand piecewise linear interpolation, including its advantages and disadvantages (convergence, stability, smoothness)
   - Understand the basic idea of how spline interpolation achieves smoothness
   - Know how a least-squares polynomial approximation is rewritten in matrix form, as a linear least-squares problem $\min_{\mathbf{a}} \|V\mathbf{a} - \mathbf{y}\|$ or as normal equations $V^T V\mathbf{a} = V^T\mathbf{y}$. Implement least-squares polynomial approximation.

3. Numerical Differentiation and Integration

   - Implement a finite difference formula. Determine its order.
   - Construct a finite difference formula for approximating first and second derivatives by polynomial interpolation.
   - Understand how roundoff error affects the accuracy of finite difference formulas when choosing h too small.
   - Know the general form of a quadrature formula (Definition 3.4) and how to derive a composite quadrature formula from it. Know the midpoint, trapezoidal, and Simpson formulas.

- Construct a quadrature formula by polynomial interpolation.
- Implement a composite quadrature formula.
- Determine the degree of exactness for a given quadrature formula. Know the connection between degree of exactness and order of a quadrature formula (Theorem 3.3 without the precise constants).

4. Linear systems – direct methods

   - Understand and implement forward/backward substitution for solving a triangular linear system.
   - Compute LU decomposition (with and without pivoting) by hand. Understand how LU decomposition is used to solve a linear system.
   - Understand why the computation of an LU decomposition is cheap for a banded matrix.
   - Understand the connection between the accuracy of the solution of a linear system and the condition number of a matrix (Lemma 4.3 without its derivation). Compute the condition number and norm of a matrix in Python.

5. Linear systems – iterative methods

   - Know the general idea of Richardson methods. Understand and implement Jacobi / Gauss-Seidel methods.
   - Know and understand Theorem 5.2 on convergence of a Richardson method.
   - Understand the connection between residual norm $\|\mathbf{r}^{(k)}\|$ and accuracy of solution (formula on Page 94).
   - Know and implement the gradient method (5.13), including the formula for the optimal steplength α_k. Know how to incorporate a preconditioner into the gradient method and why this can be helpful.

6. Ordinary differential equations

   - Know the general form (6.2) of a system of ODEs. Know how to rewrite a second-order ODE (involving first and second derivatives) in the form (6.2).
   - Know and implement the explicit Euler and Heun methods. Know the implicit Euler and Crank Nicolson methods.

- Determine the order of truncation error of a one-step method and know its implication on convergence (P. 110, top).
- Determine conditions on the step size Δt such that a given one-step method applied to the scalar model problem (6.15) is stable. Know the concept of unconditional absolute stability and its implications.

7. Boundary value problems in one dimension

   - Discretize a stationary boundary value problem with Dirichlet or Neumann boundary conditions using finite differences.
   - Implement the discretization and the solution of the discretized problem.
   - Determine the local truncation error of a discretization (Definition 7.2).