

## Numerical Analysis

Douglas Faires, Youngstown State University, (Chair, 2012-2013)  
Elizabeth Yanik, Emporia State University, (Chair, 2013-2015)  
Graeme Fairweather, Executive Editor, Mathematical Reviews, AMS  
Tim Sauer, George Mason University.

**Introduction.** Mathematical approximations have been used since ancient times to estimate solutions, but with the rise of digital computing the field of numerical analysis has become a discipline in its own right. Numerical analysts develop and study algorithms that provide approximate solutions to various types of numerical problems, and they analyze the accuracy, efficiency and robustness of these algorithms. As technology becomes ever more essential for the study of mathematics, learning algorithms that provide approximate solutions to mathematical problems and understanding the accuracy of such approximations becomes increasingly important. This report contains a description of the typical topics covered in a two-semester sequence in Numerical Analysis.

**Student Audience.** Many mathematics departments offer a two class sequence of Numerical Analysis courses. The sequence could have a large range of possible student audiences. At some universities, the first course is designed to introduce sophomores to some basic numerical algorithms in the context of learning a computer language. At others, elementary computer language skills are assumed. In this case, the sequence may focus on applications in science and engineering and be appropriate for majors in the physical sciences and engineering, as well as mathematics. A third approach would be a sequence explicitly designed for mathematics majors interested in applied areas of mathematics. In the latter cases, the topics in the two courses of the sequence may be separated by the difficulty of the mathematics e.g., numerical methods for partial differential equations may come later, giving students a better chance to acquire background from a separate theoretical course. Certainly, mixtures of all three approaches exist currently, adapted to the curriculum needs of STEM majors at individual universities.

**Prerequisites.** Students taking coursework in numerical analysis should have completed a one year university level sequence in calculus, including the study of infinite series. Some knowledge of differential equations and linear algebra is of value, and in many curricula these courses are required as a prerequisite or co-requisite. Requiring a course in elementary computer programming is also common.

**Technology.** Opinions on the amount and level of computer programming that is appropriate in a beginning Numerical Analysis course vary widely. It is possible to teach the basic ideas without any programming, although it can be argued that it is difficult to understand the real effects of using floating point arithmetic, for example, without firsthand experience on a computer. Most classes are taught using either a computing language (C, Java, Python etc.) or with a high-level package such as Maple, Mathematica, or Matlab. Each of these alternatives is quite sufficient, and the language is often determined by the preferences of the engineering/science faculty at the

institution. Choosing between a course in which the students are expected to carry out low level coding in some computer language, and a course based on a software package, which allows the instructor to take a more high level conceptual view, in the end comes down to a choice of pedagogical philosophy: Do students need the hands-on coding experience to learn the concepts, or will the concepts be clearer with some inessential details automated? Both views are valid; the decision should depend on the makeup of the students and the class goals.

**Modes of Delivery.** The first course can be effectively taught in either a small class or large lecture environment. In either case, it is important that there be ample opportunity for students to have personal contact with someone knowledgeable in the subject. Especially in a large lecture class, presentation of numerical methods should be designed to avoid excessive calculations, as this tends to distract from the overall understanding of underlying ideas. It is extremely useful to have a computer projection system in the classroom for demonstration purposes. Depending on the style of instruction, it may be advantageous for students to have the ability to compute during class or to have a computer classroom available for computer laboratory sessions. As more students acquire laptops, student access to computers in the classroom becomes less of an issue.

### **Cognitive goals**

**Analytical and critical thinking.** This is an integral part of any Numerical Analysis course. When a problem calls for an approximate solution, students need to be able to choose the most effective technique to obtain that approximate solution to a specified degree of accuracy.

**Problem-solving.** Numerical Analysis is likely to be one of the most intensive problem-solving courses an undergraduate student will take.

**Creativity.** Students are continually given opportunities to decide how to modify given techniques in order to solve problems that are similar and to modify algorithms to suit other situations. Most courses require some program or algorithm construction that requires considerable creativity on the part of the students.

**Curiosity.** The course should include an examination of problems that do not have the accuracy expected for a given method. This naturally leads to examining why methods might fail. It should also include an examination of algorithm or program construction to see how better results can be obtained with minimal computational effort and maximum accuracy. These topics and numerous others require the students to be curious, and at times somewhat ingenious.

**Communication.** Numerical analysis is a natural discipline for teamwork on the part of the students. Many classes require that solutions to assigned problems be written up independently, but suggest that students solve the problems together. Other classes assign problems to be worked on and written up in teams. Both of these approaches help improve the communication skills of the students.

## Course coverage

The following describe standard topics covered in a Numerical Analysis course or sequence. The topics after the “preliminaries” listed in the description for Numerical Analysis I are somewhat independent so there is considerable movement of topics between a first and second course, depending on the preparation of the students. For instance, if a Differential Equations course is a prerequisite for Numerical Analysis I, then approximations to differential equations are commonly studied in the first course. However, if students do not have this background, then the first course might omit this topic and include elementary techniques for solving linear systems, instead. The suggested time frames are only approximate, since time allotted for the topics in both Numerical Analysis I and II could vary significantly, depending on the amount of time allotted to student problem-solving and projects.

### Numerical Analysis I

*Preliminaries:* (2 weeks) brief review of calculus material needed for the course; discussion of various types of computational error and ways to minimize it; software that is valuable for high accuracy approximations.

*Solutions of a single nonlinear algebraic equation:* (3-4 weeks) at a minimum, the bisection, secant and Newton's methods, with an analysis of the strengths and weaknesses of each; quadratic and linear convergence methods, and special techniques for improving convergence are often included.

*Interpolation:* (3 weeks) techniques for constructing polynomials and piecewise polynomials that agree with data or functional values at specified points---these include Lagrange polynomials in numerous forms with a discussion of error analysis; classical divided-difference methods are discussed as well as methods that approximate a function and its derivative(s) at various points; Hermite polynomials, cubic splines, and other interpolation techniques are frequently covered at this time or might be postponed to a second course.

*Numerical differentiation and integration:* (3 weeks) divided-difference methods are used to approximate derivatives of functions, and stability problems with these methods are considered; various numerical integration techniques are presented from the classical Newton-Cotes and Gaussian quadrature methods to the more sophisticated extrapolation and adaptive methods; multiple integration and improper integration might also be discussed.

*Ordinary differential equations:* (3-4 Weeks) first order initial-value problems and some theory concerning unique solutions might be discussed; classical methods based on Taylor series include Euler's method and, perhaps, the midpoint or modified Euler methods; Runge-Kutta methods are explored, with particular emphasis on the Runge-Kutta method of order 4; classical predictor-corrector methods based on the Adams techniques; if time permits, more sophisticated methods based on extrapolation, and/or variable-step size methods and adaptive methods with

error control can be included; the course might also include approximations to systems of initial-value problems and the consideration of special techniques for solving stiff systems.

The following is an alternative to treating numerical methods for ordinary differential equations.

*Direct methods for linear systems:* (3-4 Weeks) Gaussian elimination and the pivoting strategies needed to control the growth of round-off error; factoring matrices to reduce computation and the connection of factoring with Gaussian elimination; factoring techniques for special classes of matrices (for example, positive definite, strictly diagonally dominant, banded) that occur in important applications; introduction of common norms for vectors to measure accuracy of approximations.

## **Numerical Analysis II**

Depending on what was covered in Numerical Analysis I, either ordinary differential equations or direct methods for linear systems: (3-4 Weeks)

*Iterative methods for linear systems:* (2-3 weeks) eigenvalues, eigenvectors, and norms of matrices; connection of these concepts to the convergence of iterative techniques; classical methods of Jacobi and Gauss-Siedel and their matrix representations; more modern methods such as the successive over-relaxation (SOR) method and the special problems posed by large, sparse matrices; condition numbers of matrices and their connection to the reliability of results obtained from approximation routines; other modern techniques such as the conjugate gradient method with preconditioning might be included.

*Eigenvalue and eigenvector approximations:* (2-3 weeks) introduction to or review of the theory from linear algebra that is required for numerical linear algebra; the power method (and its derivatives) used to determine a single specific eigenvalue and eigenvector of a matrix; The QR or similar method for determining all the eigenvalues of a matrix; the singular value decomposition (SVD) of a general matrix and some of its many applications may also be explored.

Depending on the emphasis and goals of the course, the instructor has a number of different choices for the final weeks of the course. Here are some possibilities:

*Approximation Theory:* (2-3 weeks) discrete and continuous data and function approximation, and special sets of orthogonal polynomials such as Legendre, Laguerre, and Chebyshev; the rational function approximation of Padé and its extensions; continuous and discrete trigonometric approximation of large data sets with periodic characteristics, and the application of fast Fourier transforms (FFT).

*Nonlinear Systems of Equations:* (2-3 weeks) generalization of the classical Newton's method for a single equation to a method for solving linear systems; modifications of Newton's method to reduce computation using methods derived for the classical secant method; method of steepest

descent or other techniques for obtaining starting values for more rapidly converging techniques; other modern techniques such as continuation methods could be discussed if time permits.

*Boundary Value Problems and Partial Differential Equations:* (3-4 weeks) shooting and finite difference methods for linear and nonlinear boundary value problems; finite difference techniques for partial differential equations with discussion of the accuracy difficulties for explicit difference methods; an introduction to the more modern techniques such as collocation and the finite-element method.

**Possible textbook choices:**

*Remark: The presence of a text on this list is not meant to imply an endorsement of that text, nor is the absence of a particular text from the list meant to be an anti-endorsement. The texts are chosen to illustrate the sorts of texts that support various types of transitions courses. Please note that some of the books listed were written by the authors of this report.*

The five most commonly used texts used for Numerical Analysis courses are listed below:

Burden, R. L. and J. D. Faires, *Numerical Analysis*, 9th Ed., Brooks/Cole, Boston MA, 2011.

Sauer, T. D. *Numerical Analysis*, 2<sup>nd</sup> Ed., Pearson Education, Boston MA, 2012.

Cheney, E. W. and D. R. Kincaid, *Numerical Methods and Computing*, 6<sup>th</sup> Ed., Brooks/Cole, Boston, MA 2008.

Atkinson, K. E., *An Introduction to Numerical Analysis*, 2<sup>nd</sup> Ed., John Wiley & Sons, New York NY.

Heath, M. T., *Scientific Computing, an Introductory Survey*, 2<sup>nd</sup> Ed., McGraw-Hill, New York NY, 2002.