Numerical Analysis

Kim, Hyun-Min

Department of Mathematics, Pusan National University
E-mail:hyunmin@pusan.ac.kr
Phone: 510-2596, 010-3833-8200
Office: 612

Schedule for Mid-term Exam

- Introduction to Numerical Analysis
  - What is the numerical analysis?
  - Why?
  - How?
- Roundoff Errors, Arithmetic and Algorithm
- Solutions of Equations of One Variable
  - The Bisection Method
  - The Secant Method
  - Newton's Method
  - Müller's Method
- Interpolation and Spline
  - Lagrange Polynomials
  - Divided Differences
  - Hermit Interpolation
  - Spline

- Numerical Integration and Differentiation
- Numerical Solution of Initial-Value Problems
  - Taylor Methods
  - Runge-Kutta Methods
  - Predictor-Corrector Methods
  - Extrapolation Methods
  - Adaptive Techniques
  - Methods for Systems of Equations
  - Stiff Differential Equations
- Direct Methods for Solving Linear Systems
  - Gaussian Elimination
  - Pivoting
  - Inversion
  - Factorizations
  - Special Matrices
- Mid-term Exam

Schedule for Final Term Exam

- Iterative Methods for Solving Linear Systems
  - Norms of Vectors and Matrices
  - Iterative Methods
  - Iterative Methods for Special Matrices
- Approximations
- Decompositions
  - Diagonalization
  - Jordan Canonical Form
  - Schur Decompositions
  - Singular Value Decomposition
- Approximating Eigenvalues
  - Isolation Eigenvalues
  - The Power Method
  - Householder's Method
  - The QR Method

- Solutions of Systems of Nonlinear Equations
  - Newton's Method
  - Quasi-Newton Methods
  - The Steepest Descent Method
  - Homotopy and Continuation Methods
- Boundary-Value Problems for Ordinary Differential Equations
  - The Linear Shooting Method
  - Linear Finite Difference Methods
  - The Nonlinear Shooting Method
  - Nonlinear Finite Difference Methods
- Numerical Methods for Partial Differential Equations
  - Finite Difference Methods
  - Finite Element Methods
- Final Term Exam

Linear Algebra

Math.

Linear Algebra by S. Friedberg, A. Insel and L. Spence
Chapters 1, 2, 3, 4, 5, 6, 7

Linear Algebra

Math.

Linear Algebra by S. Friedberg, A. Insel and L. Spence
Chapters 1, 2, 3, 4, 5, 6, 7

Eng.

Elementary Linear Algebra by H. Anton and C. Rorres
Chapters 1, 2, 3, 4, 5, 6, 7, 8

Calculus

Math. Calculus I, II

Calculus

Math. Calculus I, II

Eng. Calculus I, II

Differential Equations

Math.

Differential Equations

Math.

Eng. Advance Engineering Mathematics by Kreyszig

Mathematical Programming

Math.

Mathematica, Maple, MATLAB, Pythen

Mathematical Programming

Math.

Mathematica, Maple, MATLAB, Pythen

Eng.

Fortran, C, MATLAB, Pythen

Numerical Analysis

- What is the mathematics?
- What is the numerical analysis?

Definition

- The study of quantitative approximations to the solutions of mathematical problems including consideration of the errors and bounds to the errors involved.
  - Webster's New Collegiate Dictionary (1973)
- The study of methods of approximation and their accuracy, etc.
  - Chambers 20th Century Dictionary (1983)
- The branch of mathematics concerned with the development and use of numerical methods for solving problems
  - Concise Oxford Dictionary 10th Edition (1999)

New Definition

- Numerical analysis is the study of rounding errors. bad one

New Definition

- Numerical analysis is the study of rounding errors. bad one
- Numerical analysis is the study of algorithms for the problems of continuous mathematics. good one
  - Lloyd N. Trefethen (1993)

Errors

Definition

x: the true value

x^*: an approximation to x

- $|x - x^*|$: Absolute Error
- $\dfrac{|x - x^*|}{|x|}$ $(x \neq 0)$: Relative Error

Errors

Definition

x: the true value

x^*: an approximation to x

- $|x - x^*|$: Absolute Error
- $\dfrac{|x - x^*|}{|x|}$ $(x \neq 0)$: Relative Error

Which one is better?

Errors

Definition

x: the true value

x^*: an approximation to x

- $|x - x^*|$: Absolute Error
- $\dfrac{|x - x^*|}{|x|}$ $(x \neq 0)$: Relative Error

Which one is better?

Why?

Sources of Errors

- Errors in mathematical modeling: Simplifying and Assumptions

Sources of Errors

- Errors in mathematical modeling: Simplifying and Assumptions
- Blunders

Sources of Errors

- Errors in mathematical modeling: Simplifying and Assumptions
- Blunders (Programming Errors): Large programmes, Subprograms

Sources of Errors

- Errors in mathematical modeling: Simplifying and Assumptions
- Blunders (Programming Errors): Large programmes, Subprograms
- Errors in input: Errors in data transfer, uncertainties associated with measurements

Sources of Errors

- Errors in mathematical modeling: Simplifying and Assumptions
- Blunders (Programming Errors): Large programmes, Subprograms
- Errors in input: Errors in data transfer, uncertainties associated with measurements
- Machine errors by computer (Floating point arithmetic): Rounding and Chopping, Underflow and Overflow

Arithmetic

In 1985 IEEE(Institute for Electrical and Electronic Engineers) report: Binary Floating Point Arithmetic Standard 754.

Arithmetic

In 1985 IEEE(Institute for Electrical and Electronic Engineers) report: Binary Floating Point Arithmetic Standard 754.

Single, Double and Extended Precisions

Algorithms

Examining approximation procedures involving finite sequence of calculations.

- Stable: Small changes in the initial data

Algorithms

Examining approximation procedures involving finite sequence of calculations.

- Stable: Small changes in the initial data
- Unstable: Otherwise

Algorithms

Examining approximation procedures involving finite sequence of calculations.

- Stable: Small changes in the initial data
- Unstable: Otherwise
- Conditionally Stable: Stable only for certain of initial data

Convergence

$\{\alpha_n\}_{n=1}^{\infty}$, $\{\beta_n\}_{n=1}^{\infty}$: sequences

$$\lim_{n\to\infty} \beta_n = 0, \qquad \lim_{n\to\infty} \alpha_n = \alpha.$$

If $\exists\ K > 0$ s. t. $|\alpha_n - \alpha| \leq |\beta_n|$ for large n then $\{\alpha_n\}_{n=1}^{\infty}$ converges to α with the rate of convergence $O(\beta_n)$.

$$\alpha_n = \alpha + O(\beta_n)$$

Solving Nonlinear Equations

Kim, Hyun-Min

Department of Mathematics, Pusan National University

Main Theorem

Intermediate Value Theorem If $f(x)$ is continuous on $[a, b]$ and $K \in (f(a), f(b)$ or $(f(b), f(a)$
$\Longrightarrow \exists\ c \in (a, b)$ s. t. $f(c) = p$.

Main Theorem

Intermediate Value Theorem If $f(x)$ is continuous on $[a, b]$ and $K \in (f(a), f(b)$ or $(f(b), f(a)$
$\Longrightarrow \exists\, c \in (a, b)$ s. t. $f(c) = p$.

Theorem $f(x)$ is continuous on $[a, b]$ and $f(a) \cdot f(b) < 0$
$\Longrightarrow \exists\, c \in (a, b)$ s. t. $f(c) = 0$.

Main Theorem

Intermediate Value Theorem If $f(x)$ is continuous on $[a, b]$ and $K \in (f(a), f(b)$ or $(f(b), f(a)$
$\Longrightarrow \exists\, c \in (a, b)$ s. t. $f(c) = p$.

Theorem $f(x)$ is continuous on $[a, b]$ and $f(a) \cdot f(b) < 0$
$\Longrightarrow \exists\, c \in (a, b)$ s. t. $f(c) = 0$.

The Nested Interval Property If $I_n = [a_n, b_n]$, $n \in \mathbb{N}$ is a nested closed bounded intervals then $\exists\, \xi \in \mathbb{R}$ s. t. $\xi \in I_n$ for all $n \in \mathbb{N}$.

Algorithm

Input data: $y = f(x)$, a, b,

Algorithm

Input data: $y = f(x)$, a, b, tolerance,

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

If $f(p_k) = 0$ then $c = p_k$

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

If $f(p_k) = 0$ then $c = p_k$

If $f(p_k) \neq 0$ then

$$\begin{cases} f(p_k) \cdot f(a_k) < 0 & \Longrightarrow \quad a_{k+1} = a_k \quad \text{and} \quad b_{k+1} = p_k \\ f(p_k) \cdot f(b_k) < 0 & \Longrightarrow \quad a_{k+1} = p_k \quad \text{and} \quad b_{k+1} = b_k \end{cases}$$

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

If $f(p_k) = 0$ then $c = p_k$

If $f(p_k) \neq 0$ then

$$\begin{cases} f(p_k) \cdot f(a_k) < 0 \implies a_{k+1} = a_k \quad \text{and} \quad b_{k+1} = p_k \\ f(p_k) \cdot f(b_k) < 0 \implies a_{k+1} = p_k \quad \text{and} \quad b_{k+1} = b_k \end{cases}$$

Stop

- $|b_n - a_n| <$ tolerance

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

If $f(p_k) = 0$ then $c = p_k$

If $f(p_k) \neq 0$ then

$$\begin{cases} f(p_k) \cdot f(a_k) < 0 & \Longrightarrow \quad a_{k+1} = a_k \quad \text{and} \quad b_{k+1} = p_k \\ f(p_k) \cdot f(b_k) < 0 & \Longrightarrow \quad a_{k+1} = p_k \quad \text{and} \quad b_{k+1} = b_k \end{cases}$$

Stop

- $|b_n - a_n| <$ tolerance
- $|f(p_n)| <$ tolerance

Algorithm

Input data: $y = f(x)$, a, b, tolerance, maximum number of iterations

Set: $a_1 = a$, $b_1 = b$ and $p_1 = \frac{a_1+b_1}{2}$

If $f(p_k) = 0$ then $c = p_k$

If $f(p_k) \neq 0$ then

$$\begin{cases} f(p_k) \cdot f(a_k) < 0 & \Longrightarrow & a_{k+1} = a_k & \text{and} & b_{k+1} = p_k \\ f(p_k) \cdot f(b_k) < 0 & \Longrightarrow & a_{k+1} = p_k & \text{and} & b_{k+1} = b_k \end{cases}$$

Stop

- $|b_n - a_n| <$ tolerance
- $|f(p_n)| <$ tolerance
- $|\frac{f(p_n)}{p_n}| <$ tolerance