MA579H Scientific Computing

Lecture 1: Introduction

Rafikul Alam
Department of Mathematics
IIT Guwahati

Course Logistics

Instructor: Rafikul Alam

Lab Teaching Assistant: Dr. Prince Kanhya

Course Logistics

Instructor: Rafikul Alam

Lab Teaching Assistant: Dr. Prince Kanhya

I will post lecture slides, homework and programming assignments on Microsoft Teams.

Course Syllabus

Definition and sources of errors; Solutions of nonlinear equations; Bisection method, Newton's method and its variants, fixed point iterations, convergence analysis; Newton's method for non-linear systems;

Course Syllabus

Definition and sources of errors; Solutions of nonlinear equations; Bisection method, Newton's method and its variants, fixed point iterations, convergence analysis; Newton's method for non-linear systems;

Finite differences, polynomial interpolation; Numerical integration - Trapezoidal and Simpson's rules, Gaussian quadrature;

Course Syllabus

Definition and sources of errors; Solutions of nonlinear equations; Bisection method, Newton's method and its variants, fixed point iterations, convergence analysis; Newton's method for non-linear systems;

Finite differences, polynomial interpolation; Numerical integration - Trapezoidal and Simpson's rules, Gaussian quadrature;

Initial value problems - Taylor series method, Euler and modified Euler methods, Runge-Kutta methods.

Course Syllabus

Definition and sources of errors; Solutions of nonlinear equations; Bisection method, Newton's method and its variants, fixed point iterations, convergence analysis; Newton's method for non-linear systems;

Finite differences, polynomial interpolation; Numerical integration - Trapezoidal and Simpson's rules, Gaussian quadrature;

Initial value problems - Taylor series method, Euler and modified Euler methods, Runge-Kutta methods.

Textbooks:

- D. Kincaid and W. Cheney, Numerical Mathematics and Computing, 7th Edn., Cengage, 2013.
- K. E. Atkinson, Introduction to Numerical Analysis, 2nd Edn., John Wiley, 1989.
- Timothy Sauer, Numerical Analysis, 3rd Edition, Pearson, 2018.

Assessment

Grading will be based on continuous assessments.

- Quiz: 30 %
- End/Mid semester examination: 60 %
- Assignment/Viva: 10 %

Lab experiments will be assessed as a part of MA581 Numerical Computations Lab. Grading will be based on assignments (20 %), mid semester examination (30 %) and end semester examination (50 %).

Introduction

What is scientific computing?

Introduction

What is scientific computing? Scientific computing is a subject that deals with computer methods for numerically solving mathematical problems that arise in science and engineering.

Introduction

What is scientific computing? Scientific computing is a subject that deals with computer methods for numerically solving mathematical problems that arise in science and engineering.

By computer methods we mean

(a) design,

(b) analysis and

(c) implementation

of numerical algorithms that yield approximate solutions to mathematical problems using finite precision arithmetic on computers.

Introduction

What is scientific computing? Scientific computing is a subject that deals with computer methods for numerically solving mathematical problems that arise in science and engineering.

By computer methods we mean

(a) design,

(b) analysis and

(c) implementation

of numerical algorithms that yield approximate solutions to mathematical problems using finite precision arithmetic on computers.

Typical problems addressed in scientific computing include:

- Approximation of functions by interpolation
- Numerical differentiation and integration

Introduction

What is scientific computing? Scientific computing is a subject that deals with computer methods for numerically solving mathematical problems that arise in science and engineering.

By computer methods we mean

(a) design,

(b) analysis and

(c) implementation

of numerical algorithms that yield approximate solutions to mathematical problems using finite precision arithmetic on computers.

Typical problems addressed in scientific computing include:

- Approximation of functions by interpolation
- Numerical differentiation and integration
- Solution of ordinary and partial differential equations
- Solution of linear and nonlinear algebraic equations

Design

Design of numerical methods (numerical algorithms) is one of the most important aspects of scientific computing.

Design

Design of numerical methods (numerical algorithms) is one of the most important aspects of scientific computing.

- Algorithm design requires finding a clever way of solving a problem that is fast and yields realiable solutions.

Design

Design of numerical methods (numerical algorithms) is one of the most important aspects of scientific computing.

- Algorithm design requires finding a clever way of solving a problem that is fast and yields realiable solutions.

Problem: Solve $ax^2 + bx + c = 0$, where $a, b, c \in \mathbb{R}$

Design

Design of numerical methods (numerical algorithms) is one of the most important aspects of scientific computing.

- Algorithm design requires finding a clever way of solving a problem that is fast and yields realiable solutions.

Problem: Solve $ax^2 + bx + c = 0$, where $a, b, c \in \mathbb{R}$

Classical method: Naive use of the formula yields

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Design

Design of numerical methods (numerical algorithms) is one of the most important aspects of scientific computing.

- Algorithm design requires finding a clever way of solving a problem that is fast and yields realiable solutions.

Problem: Solve $ax^2 + bx + c = 0$, where $a, b, c \in \mathbb{R}$

Classical method: Naive use of the formula yields

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

Modified method: Clever use of the formula yields $(x_1 x_2 = c/a)$

$$x_1 = -\frac{b + \mathrm{sign}(b)\sqrt{b^2 - 4ac}}{2a}, \quad x_2 = \frac{c}{a x_1}$$

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$x_1 = \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10}$$

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$
\begin{aligned}
x_1 &= \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10} \\
x_2 &= \frac{-10^7 + \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -3.0 \times 10^{-7}
\end{aligned}
$$

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$
\begin{aligned}
x_1 &= \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10} \\
x_2 &= \frac{-10^7 + \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -3.0 \times 10^{-7}
\end{aligned}
$$

Implementing classical method in MATLAB yields

x_1 `= -1.0000e+10,` x_2 `= 0`

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$
\begin{aligned}
x_1 &= \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10} \\
x_2 &= \frac{-10^7 + \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -3.0 \times 10^{-7}
\end{aligned}
$$

Implementing classical method in MATLAB yields

x_1 `= -1.0000e+10,` x_2 `= 0`

Implementing modified method in MATLAB yields

x_1 `= -1.0000e+10,` x_2 `= -3.0000e-07`

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$x_1 = \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10}$$
$$x_2 = \frac{-10^7 + \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -3.0 \times 10^{-7}$$

Implementing classical method in MATLAB yields

x_1 = -1.0000e+10, x_2 = 0

Implementing modified method in MATLAB yields

x_1 = -1.0000e+10, x_2 = -3.0000e-07

Problem: Compute $\dfrac{e^x - 1 - x}{x^2}$ for $x \approx 0$.

Example

Solve $10^{-3}x^2 + 10^7 x + 3 = 0$

$$\begin{aligned} x_1 &= \frac{-10^7 - \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -1.0 \times 10^{10} \\ x_2 &= \frac{-10^7 + \sqrt{10^{14} - 12 \times 10^{-3}}}{2 \times 10^{-3}} \approx -3.0 \times 10^{-7} \end{aligned}$$

Implementing classical method in MATLAB yields

x_1 = -1.0000e+10, x_2 = 0

Implementing modified method in MATLAB yields

x_1 = -1.0000e+10, x_2 = -3.0000e-07

Problem: Compute $\dfrac{e^x - 1 - x}{x^2}$ for $x \approx 0$. THINK.

Analysis

Analysis of numerical methods (numerical algorithms) is the heart of the subject matter and consists of analyzing

Analysis

Analysis of numerical methods (numerical algorithms) is the heart of the subject matter and consists of analyzing

- stability (reliability) of numerical methods (algorithms)
- efficiency of numerical methods (i.e., complexity of algorithms),

Analysis

Analysis of numerical methods (numerical algorithms) is the heart of the subject matter and consists of analyzing

- stability (reliability) of numerical methods (algorithms)
- efficiency of numerical methods (i.e., complexity of algorithms),
- the effect of finite precision arithmetic upon algorithms, and
- approximation errors and accuracy of solutions.

Example 1

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be differentiable. Then $f'(x) := \lim_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$.

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x, h) := |f'(x) - D_h f(x)|$.

Example 1

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be differentiable. Then $f'(x) := \lim_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$.

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x, h) := |f'(x) - D_h f(x)|$.

Since $\lim_{h \to 0} D_h f(x) = f'(x)$, in theory,

$$|h| \text{ small } \implies E(x, h) \text{ is small.}$$

But is it true numerically?

Example 1

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be differentiable. Then $f'(x) := \lim_{h \to 0} \dfrac{f(x+h) - f(x)}{h}$.

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x, h) := |f'(x) - D_h f(x)|$.

Since $\lim_{h \to 0} D_h f(x) = f'(x)$, in theory,

$$|h| \text{ small } \implies E(x, h) \text{ is small.}$$

But is it true numerically?

For $f(x) = e^x$, we plot $E(0, h)$ as a function of h.

The plot shows that $|E(0, h)|$ decreases as h decreases to certain $h_{\min}$ and then starts increasing when $h < h_{\min}$ further decreases to 0.

Example 1

Let $f : \mathbb{R} \longrightarrow \mathbb{R}$ be differentiable. Then $f'(x) := \lim_{h\to 0} \dfrac{f(x+h) - f(x)}{h}$.

Define $D_h f(x) := \dfrac{f(x+h) - f(x)}{h}$ and $E(x, h) := |f'(x) - D_h f(x)|$.

Since $\lim_{h\to 0} D_h f(x) = f'(x)$, in theory,

$$|h| \text{ small } \implies E(x, h) \text{ is small.}$$

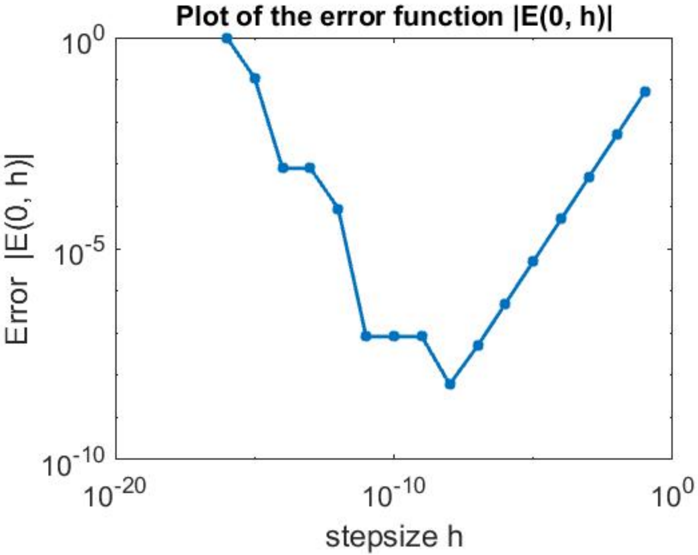
But is it true numerically?

For $f(x) = e^x$, we plot $E(0, h)$ as a function of h.

The plot shows that $|E(0, h)|$ decreases as h decreases to certain $h_{\min}$ and then starts increasing when $h < h_{\min}$ further decreases to 0.

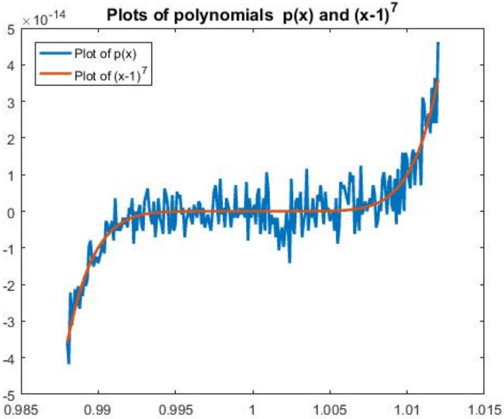
How do you explain this behavior of $|E(0, h)|$?

Example 1


Plot of the error function |E(0, h)|
Error |E(0, h)|
10^{0}
10^{-5}
10^{-10}
10^{-20}
10^{-10}
10^{0}
stepsize h

Example 2

Let $p(x) = x^7 - 7x^6 + 21x^5 - 35x^4 + 35x^3 - 21x^2 + 7x - 1 = (x-1)^7$.



Does the plot look like a polynomial? What explains this behaviour?

Implementation

The implementation of numerical methods (algorithms) deals with development of efficient, reliable and portable software for solving mathematical problems that arise in science and engineering.

Outline of topics

- Finite precision arithmetic and propagation errors
- Polynomial interpolation

Outline of topics

- Finite precision arithmetic and propagation errors
- Polynomial interpolation
- Numerical integration and differentiation
- Solution of nonlinear equations

Outline of topics

- Finite precision arithmetic and propagation errors
- Polynomial interpolation
- Numerical integration and differentiation
- Solution of nonlinear equations
- Solution of initial value ordinary differential equations

Learning outcome

This course will equip you with rigorous knowledge of the core topics, namely,

- root-finding techniques
- approximation of functions by polynomials
- numerical quadratures
- solution of initial value problem (ODEs)

using appropriate mathematical syntax and terminology.

Learning outcome

This course will equip you with rigorous knowledge of the core topics, namely,

- root-finding techniques
- approximation of functions by polynomials
- numerical quadratures
- solution of initial value problem (ODEs)

using appropriate mathematical syntax and terminology.

You will learn fundamental principles underpinning Scientific Computing and apply these principles to design algorithms for solving mathematical problems.

Learning outcome

This course will equip you with rigorous knowledge of the core topics, namely,

- root-finding techniques
- approximation of functions by polynomials
- numerical quadratures
- solution of initial value problem (ODEs)

using appropriate mathematical syntax and terminology.

You will learn fundamental principles underpinning Scientific Computing and apply these principles to design algorithms for solving mathematical problems.

Also, you will learn how to use MATLAB to implement these algorithms and adapt the codes for more general problems.

***