Foundations of Scientific Computing
CAS 708/CSE 700

Guangning (Gary) Tan

School of Computational Science and Engineering

McMaster University

tgnteach@gmail.com

September 8, 2015

Contents

Overview

Brief review

MATLAB tutorial

Introduction

- Instructor: Ned Nedialkov
- www.cas.mcmaster.ca/~nedialk/COURSES/cas708/
- Materials for SE4X03 (Sanzheng Qiao)
- Textbook: A first course in numerical methods
- Prerequisites:
  - calculus and linear algebra
  - background in numerical methods
  - basic programming in Matlab
- Link to this PDF
  www.cas.mcmaster.ca/~tang4/CSE700.pdf

Topics

- Floating point arithmetic
- Linear systems
- Linear least squares, orthogonal transformations
- Eigenvalues/vectors, singular values/vectors
- Nonlinear systems and optimization
- Polynomial interpolation
- Numerical differentiation/integration
- Initial value problems in ordinary differential equations (IVP ODEs)

Textbook: Michael Heath, Scientific computing: an introductory survey

Floating point arithmetic

- In IEEE single precision,

$$(3.2)_{10} = 0 \quad 10000000 \quad 10011001100110011001101$$

- An online converter
  http://www.h-schmidt.net/FloatConverter/IEEE754.html
- Roundoff error analysis
- Avoid cancellation

  E.g., which is more accurate, $x^2 - y^2$ or $(x - y)(x + y)$ if $|x| \approx |y|$?
- D. Goldberg, What every computer scientist should know about floating-point arithmetic

Solving system of linear equations $Ax = b$

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 5 \\ -1 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 3 \end{bmatrix}$$

- Cramer's rule? Calculating inverse of A?
- Gaussian elimination
- LU factorization
- Backward/forward substitution

Solving system of nonlinear equations

$$\mathbf{0} = \mathbf{f}(x, y) = \begin{cases} f_1 = x^2 + y^2 - 100 \\ f_2 = x + y - 14 \end{cases}$$

- Newton's (iterative) method
- Start with an initial guess (x_0, y_0)
- Jacobian is $\mathbf{J} = \begin{bmatrix} 2x & 2y \\ 1 & 1 \end{bmatrix}$
- If $\mathbf{J}$ is nonsingular at (x_i, y_i), solve

$$\begin{bmatrix} 2x_i & 2y_i \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{bmatrix} = -\mathbf{f}(x_i, y_i)$$

for (x_{i+1}, y_{i+1})

Optimization problems

$$\min_{x,y} \; 2x^2 + 3xy + 3y^2 - 2x + 4y + 6$$

- Necessary condition: solve $\nabla f(x,y) = \mathbf{0}$

$$\begin{cases} 4x + 3y - 2 = 0 \\ 3x + 6y + 4 = 0 \end{cases}$$

- In general, solve a nonlinear system for a local optimization problem
- Jacobian of this nonlinear system is called Hessian of $f(x,y)$

Ordinary differential equations

Solve $y' = -y$ and $y(0) = 1$

- Forward Euler: approximate y' by finite difference

$$y'(t_i) = \frac{y_{i+1} - y_i}{h}$$

  where $y_i = y(t_i)$ and $h = t_{i+1} - t_i$

- $y_{i+1} = y_i - hy_i = (1 - h)y_i$
- E.g., if $h = 0.05$, then $y(0.05) = (1 - 0.05) \times 1 = 0.95$
- General form is $\mathbf{y}' = \mathbf{f}(t, \mathbf{y})$ with initial values $\mathbf{y}(0) = \mathbf{y}_0$

Numerical integration

Compute $\int_0^\pi x \sin x \, \mathrm{d}x$ using trapezoid rule

- Divide $[0, \pi]$ into N equally spaced sub-intervals: $[x_{i-1}, x_i]$, where $x_i = i\pi/N$, $i = 0, 1, \ldots, N$
- Approximate each sub-area

$$\int_{x_{i-1}}^{x_i} f(x) \, \mathrm{d}x \approx \frac{h}{2} \left(f(x_{i-1}) + f(x_i) \right)$$

$$\int_0^\pi f(x) \, \mathrm{d}x \approx \frac{h}{2} \left(f(0) + 2 \sum_{i=1}^{N-1} f(x_i) + f(\pi) \right)$$

where $f(x) = x \sin x$ and $h = \pi/N$

MATLAB

- Latest release: R2015b (V8.6), September 3, 2015
- Access R2014a (V8.3) in CAS mills server
  www.cas.mcmaster.ca/~tang4/SE4TE3/matlab.html
- Objectives
  - basic MATLAB commands and functions
  - implement numerical algorithms
  - debug
  - interpret and analyze numerical results
  - optimize code
  - develop a good coding style

Hands-on materials

- MATLAB Primer by MathWorks
- Textbooks by Cleve Moler
  - Numerical Computing with MATLAB
  - Experiments with MATLAB
- MATLAB's documentations

  Simply Google "www.mathworks.com/help/pdf_doc/matlab/"
- Richard Johnson, MATLAB Style Guidelines 2.0

Simple MATLAB tutorial

- Simple commands
- Colon notation
- Built-in functions
- MATLAB's documentation
- M-files: scripts and functions
- Plots
- Debugging
- Optimize code
- Commenting