

Unit 0. Introduction

Numerical Analysis

Feb. 24, 2015

Numerical Analysis

- Mathematics are important in our study, research and work.
 - Mathematical formulation provides clear and precise description of the problem.
 - Quantified or analytically.
 - Analytic solutions have been studied throughout your lives.
 - Few problems have closed form solutions.
 - How do you solve mathematical problems to get numerical solutions?
- Numerical methods apply computation tools (computers) to solve mathematical problems to get numerical solutions.
 - Explore the mathematical properties of the problems.
 - Develop algorithm to solve the problem effectively.
 - Find approximated solutions if more efficient.
 - Overcome computer inexactness.
- Numerical methods have been used extensively in our daily lives already.
 - List your examples.

Topics to Be Studied

- In this course, we will study the following topics in numerical analysis.
 - Linear system solutions,
 - Errors,
 - Iterative solution methods,
 - Eigenvalues and eigenvectors,
 - Interpolations,
 - Integrations,
 - Nonlinear system solutions,
 - Ordinary differential equations,
 - Boundary value problems.

Linear System Solutions

- You know how to solve this problem.

$$\begin{array}{rcl} 2x_1 & +x_2 & = 1 \\ x_1 & +2x_2 & = 1 \end{array}$$

- How about this?

$$\begin{array}{rclclclcl} 70x_1 & +60x_2 & +50x_3 & +40x_4 & +30x_5 & +20x_6 & +10x_7 & = 1 \\ 60x_1 & +70x_2 & +60x_3 & +50x_4 & +40x_5 & +30x_6 & +20x_7 & = 2 \\ 50x_1 & +60x_2 & +70x_3 & +60x_4 & +50x_5 & +40x_6 & +30x_7 & = 3 \\ 40x_1 & +50x_2 & +60x_3 & +70x_4 & +60x_5 & +50x_6 & +40x_7 & = 4 \\ 30x_1 & +40x_2 & +50x_3 & +60x_4 & +70x_5 & +60x_6 & +50x_7 & = 3 \\ 20x_1 & +30x_2 & +40x_3 & +50x_4 & +60x_5 & +70x_6 & +60x_7 & = 2 \\ 19x_1 & +29x_2 & +39x_3 & +49x_4 & +59x_5 & +69x_6 & +59x_7 & = 1 \end{array}$$

- What if we have 1,000 or 1,000,000 variables?
 - Can we solve it efficiently?

Errors

- We know computer numbers are inexact.
 - Experimental data have inherent inaccuracy also.
 - How do these errors affect our solutions?
 - Can the algorithm be rewritten for smaller errors?

$$\begin{bmatrix} 70 & 60 & 50 & 40 & 30 & 20 & 10 \\ 60 & 70 & 60 & 50 & 40 & 30 & 20 \\ 50 & 60 & 70 & 60 & 50 & 40 & 30 \\ 40 & 50 & 60 & 70 & 60 & 50 & 40 \\ 30 & 40 & 50 & 60 & 70 & 60 & 50 \\ 20 & 30 & 40 & 50 & 60 & 70 & 60 \\ 19 & 29 & 39 & 49 & 59 & 69 & 59 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}.$$

$$\mathbf{x}^T = [0.45 \ 0 \ 0 \ 0.1 \ 0 \ -3.9 \ 4.35]$$

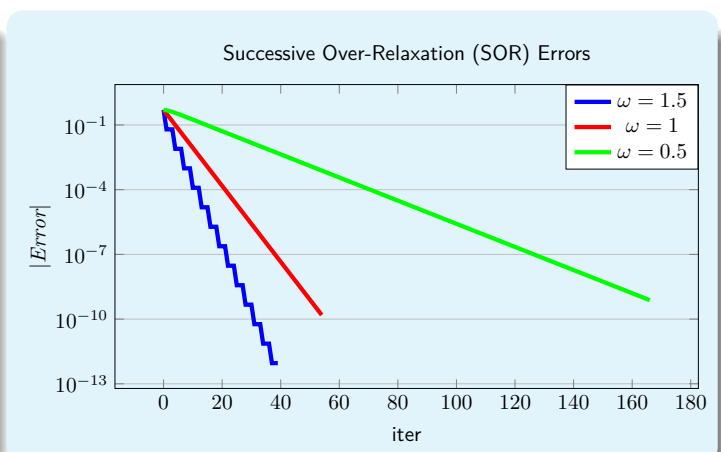
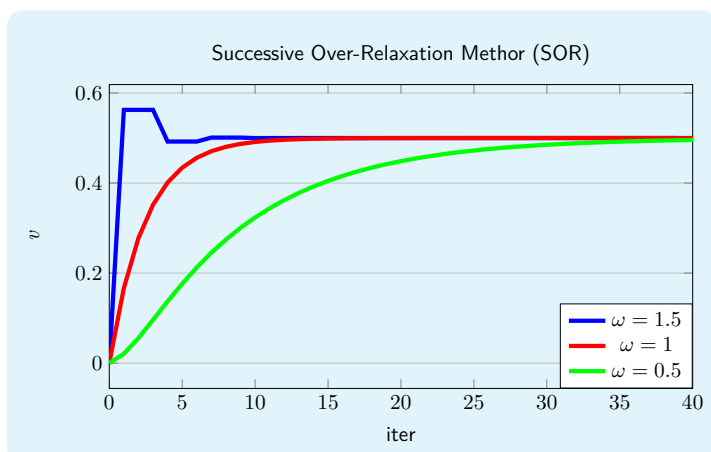
$$\begin{bmatrix} 70 & 60 & 50 & 40 & 30 & 20 & 10 \\ 60 & 70 & 60 & 50 & 40 & 30 & 20 \\ 50 & 60 & 70 & 60 & 50 & 40 & 30 \\ 40 & 50 & 60 & 70 & 60 & 50 & 40 \\ 30 & 40 & 50 & 60 & 70 & 60 & 50 \\ 20 & 30 & 40 & 50 & 60 & 70 & 60 \\ 19 & 29 & 39 & 49 & 59 & 69 & 59 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 3 \\ 2 \\ 1.1 \end{bmatrix}.$$

$$\mathbf{x}^T = [0.4 \ 0 \ 0 \ 0.1 \ 0 \ -3.5 \ 3.9]$$

- A small error can cause significant changes in solutions.

Iterative Solution Methods

- Solving a linear system with n variables takes $\mathcal{O}(n^3)$ CPU time.
- For large n the solution time can be too long.
- Iterative solution methods find an approximate solution with less amount of time.
- Matrix may need to possess some special properties to be applicable.
- More general solutions have been developed, but not covered in this course.



Eigenvalues and Eigenvectors

- You know how to find the eigenvalues and eigenvectors for

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

- How about this?

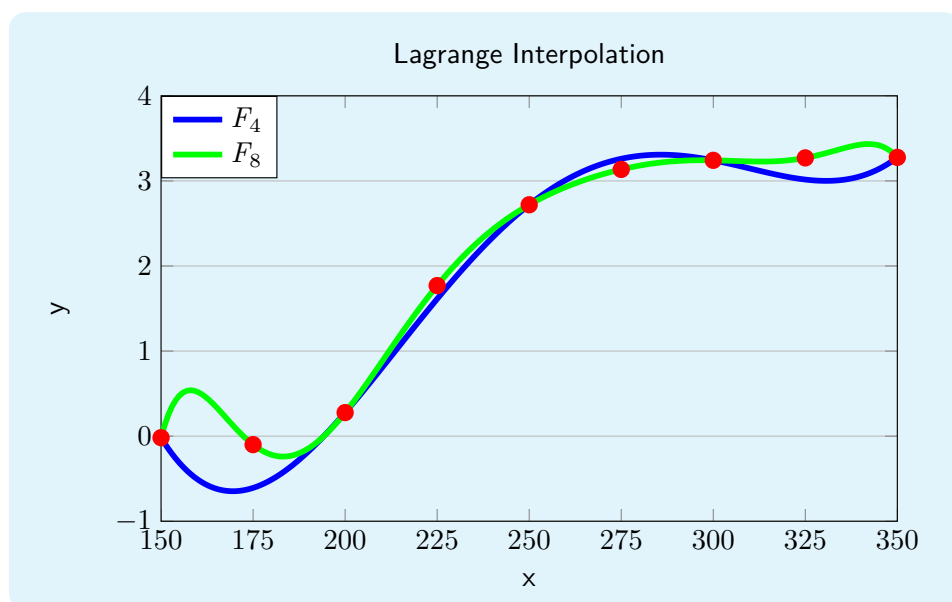
$$\begin{bmatrix} 70 & 60 & 50 & 40 & 30 & 20 & 10 \\ 60 & 70 & 60 & 50 & 40 & 30 & 20 \\ 50 & 60 & 70 & 60 & 50 & 40 & 30 \\ 40 & 50 & 60 & 70 & 60 & 50 & 40 \\ 30 & 40 & 50 & 60 & 70 & 60 & 50 \\ 20 & 30 & 40 & 50 & 60 & 70 & 60 \\ 19 & 29 & 39 & 49 & 59 & 69 & 59 \end{bmatrix}$$

$$\lambda^T = [339.07 \ 96.29 \ 20.67 \ 10.89 \ 6.80 \ 5.41 \ -0.12]$$

- What if we have a $1,000 \times 1,000$ or $1,000,000 \times 1,000,000$ matrix?
 - Can we solve it efficiently?

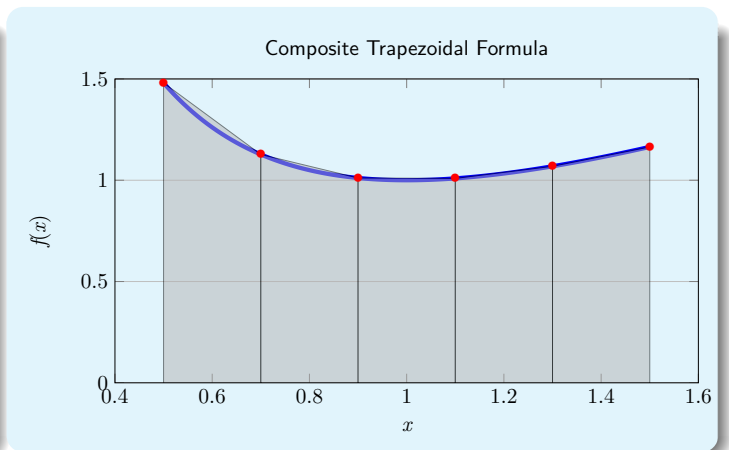
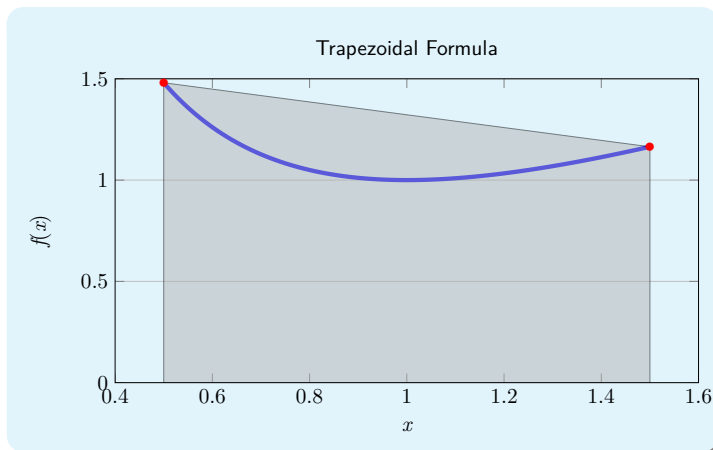
Interpolations

- Given a set of data, find a function to represent these data points.
 - Polynomial interpolations,
 - Rational interpolations (not covered),
 - Spline interpolations.



Integrations

- Integral equations have few closed form solutions.
- Numerical integration has been studied and adopted early.



Nonlinear System Solutions

- Example of nonlinear function

$$f(x) = \log^2(x) - x + 0.9$$

- What is the value of x such that $f(x) = 0$?
 - $f(x)$ can be any nonlinear function.
- What are the roots of

$$x^7 + 6x^5 + 3x^4 + 10x^3 + 25x^2 + 100 = 0.$$

- It has been proven that no closed form solution exists for polynomial of order greater than 5.
 - Numerical methods are usually necessary to find the solutions.

Ordinary Differential Equations

- Differential equations have been used extensively in science and engineering fields.

$$\begin{aligned}\frac{dx(t)}{dt} &= f(t, x(t)), & t \geq 0, \\ x(0) &= x_0.\end{aligned}$$

- Finding the solution quickly and accurately is the key of this study.
- Electric circuits with capacitors or inductors, or both, need to be solved using this method if time domain solution is needed.
- Again, solution is usually an approximation.
 - Limitations and accuracy should be well versed to get acceptable solutions.
- In many applications, $f(t, x(t))$ is nonlinear. Thus, solving ODE needs to apply all techniques learned in this course.

Practice and Programming

- We will analyze different numerical analysis topics and develop algorithms or methods to solving these problems.
- These algorithms or methods can be implemented using different tools.
- Some tools are primitive and need more efforts.
- Some with built-in functions and are very easy to use.
- In this class, we will adopt C++ to practice detailed implementations.
- Basic functions will be developed such that you know all the basics.
- Using [operator overloading](#), our program will be very easy to understand and minimize potential errors.
- [C++ class](#) and [operator overloading](#) will be reviewed next.
- Since all the algorithms will be discussed, the programming difficulty level is actually quite low.
- You should enjoy our homework every week.