Math 541 - Numerical Analysis

Lecture Notes – Introduction to Numerical Analysis

Joseph M. Mahaffy,
⟨jmahaffy@mail.sdsu.edu⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://jmahaffy.sdsu.edu

Spring 2018

SDSU

Outline

Contact Information

Professor Joseph Mahaffy

Office	GMCS-593
Email	jmahaffy@mail.sdsu.edu
Web	http://jmahaffy.sdsu.edu
Phone	(619)594-3743
Office Hours	MW: 12:00-13:50 in MLC and by appointment

Basic Information: Text/Topics

Text:

Cleve Moler: Numerical Computing in Matlab

1. Mathematical Preliminaries – Taylor's series
2. MatLab Basics
3. Error Analysis
4. Zeros of Functions
5. Numerical Integration – Quadrature
6. Numerical Linear Algebra
7. Interpolation – Splines
8. Least Squares

Other Numerical Analysis Courses

- **Math 542:** ***Numerical Solutions of Differential Equations***
  - Initial-Value Problems for ODEs
  - Boundary Value Problems for ODEs
- **Math 543:** ***Numerical Matrix Analysis***
  - Iterative Techniques in Matrix Algebra
  - Approximating Eigenvalues
- **Math 693A:** ***Advanced Numerical Analysis (Numerical Optimization)***
  - Numerical Solution of Nonlinear Systems of Equations
- **Math 693B:** ***Advanced Numerical Analysis (Numerics for PDEs)***
  - Numerical Solution of PDEs

Basic Information: Grading

Approximate Grading

Homework, including WeBWorK*	45%
Lab Report+	5%
Exams and Final×	50%

* Both theoretical and implementation (programming) — MatLab will be the primary programming language.

\+ Formal Lab Reports will be written on several applied problems.

× Likely to be 2 Midterms and Final with part being Take-home. Final: Monday, May 7, 10:30 – 12:30.

Expectations and Procedures, I

- Most class attendance is OPTIONAL — Homework and announcements will be posted on the class web page.
If/when you attend class:
    - Please be on time.
    - Please pay attention.
    - Please turn off mobile phones.
    - Please be courteous to other students and the instructor.
    - Abide by university statutes, and all applicable local, state, and federal laws.

Expectations and Procedures, II

- Please, turn in assignments on time. (The instructor reserves the right not to accept late assignments.)
- The instructor will make special arrangements for students with documented learning disabilities and will `try` to make accommodations for other unforeseen circumstances, *e.g.* illness, personal/family crises, etc. in a way that is fair to all students enrolled in the class. ***Please contact the instructor EARLY regarding special circumstances.***
- Students are expected ***and encouraged*** to ask questions in class!
- Students are expected ***and encouraged*** to to make use of office hours! If you cannot make it to the scheduled office hours: contact the instructor to schedule an appointment!

Expectations and Procedures, III

- Missed midterm exams: Don't miss exams! The instructor reserves the right to schedule make-up exams, make such exams oral presentation, and/or base the grade solely on other work (including the final exam).
- Missed final exam: Don't miss the final! Contact the instructor ASAP or a grade of incomplete or F will be assigned.
- ***Academic honesty***: Submit your own work. Any cheating will be reported to University authorities and a **ZERO** will be given for that HW assignment or Exam.

MatLab

- Students can obtain **MatLab** from ROHAN Academic Computing.
- Google **SDSU MatLab** or access **http://www-rohan.sdsu.edu/~download/matlab.html**.
- **MatLab** and **Maple** can also be accessed in the **Computer Labs GMCS 421**, **422**, and **425**.
- You may also want to consider buying the student version of MatLab: **http://www.mathworks.com/**

Math 541: Formal Prerequisites

`Math 254` or `Math 342A` or `AE 280`

- These courses all have sections on basic **Linear Algebra** and assume knowledge of **Calculus** (especially ***Taylor's Theorem***)

`CS 107` or `Math 242`

- These courses introduce basic **Computer Programming**

Math 541: Introduction — What we will learn

1. Numerical tools for problem solving
2. How to translate mathematical problems into **MatLab** code
3. Error and convergence analysis
   - Computational mathematics has errors
   - Must understand sources of errors and improvement of algorithms
4. How to implement **Calculus** on computers: Solve $f(x) = 0$, Integration, ...
5. Use **MatLab** to solve problems in **Linear Algebra**
6. Work with data: Fitting with splines and least squares best fits

Math 541: Introduction — Why???

Q: Why are numerical methods needed?

A: To accurately approximate the solutions of problems that cannot be solved exactly.

Q: What kind of applications can benefit from numerical studies?

A: Engineering, physics, chemistry, computer, biological and social sciences.

Image processing / computer vision, computer graphics (rendering, animation), climate modeling, weather predictions, "virtual" crash-testing of cars, medical imaging (CT = Computed Tomography), AIDS research (virus decay vs. medication), financial math...

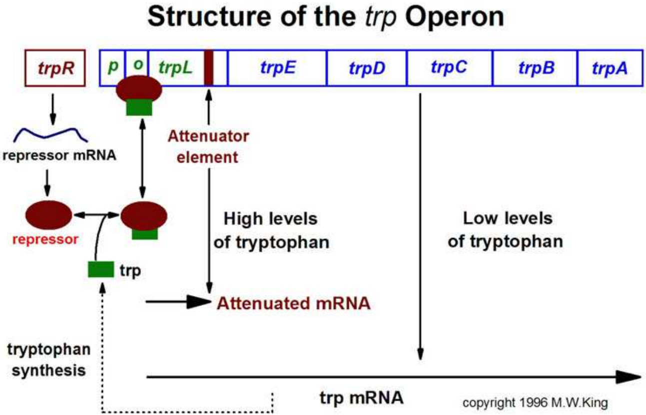
Math 541: Introduction — Computing Efficiency

Numerical tools for problem solving:

- Computers are getting faster, but the computer's speed is only one (a big one for sure!) part of the overall performance for a computation...
  - Computing speed depends on **FLOPS** (floating-point operations or number of additions and multiplications) and ***memory accesses***. These are largely questions of computer architecture and won't be examined in this course much.
  - Numerical Algorithms are the center of this course, and their efficiency affects performance.

Research Problem from my Work

Genetic Control by Repression



Model for Conrol by Repression

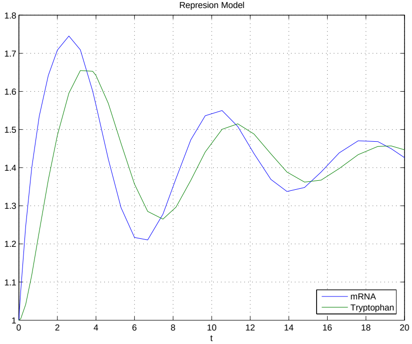
- $x_1(t)$ is the concentration of mRNA
- $x_2(t)$ is the concentration of the tryptophan (endproduct)
- **Endproduct inhibition** or a **negative feedback system** can result in oscillatory behavior
- System of first order delay differential equations (DDE):

$$\begin{aligned} \frac{dx_1(t)}{dt} &= \frac{a_1}{1 + kx_2^n(t-R)} - b_1x_1(t) \\ \frac{dx_2(t)}{dt} &= a_2x_1(t) - b_2x_2(t) \end{aligned}$$

- Solve numerically, such as **MatLab's** *dde23.m* delay differential equation solver

Simulation of Repression Model

With $a_1 = 2$, $a_2 = b_1 = b_2 = 1$, $n = 4$, and $R = 2$, the model is simulated using **MatLab's** *dde23.m*



Math 542 studies the Runge-Kutta-Felberg method for numerically integrating ordinary differential equations, a related method

MatLab code available from Website.

Equilibrium Analysis

- Qualitative analysis of a differential equation begins by finding all equilibria
- Equilibria solve the derivatives equal to zero

$$\begin{aligned} \frac{a_1}{1+k\bar{x}_2^n} - b_1\bar{x}_1 &= 0 \\ a_2\bar{x}_1 - b_2\bar{x}_2 &= 0 \end{aligned}$$

- This is a system of nonlinear equations equal to zero
- This easily reduces to a nonlinear scalar equation,

$$\frac{a_1}{1+k\bar{x}_2^n} - \frac{b_1 b_2}{a_2}\bar{x}_2 = 0 \quad \text{with} \quad \bar{x}_1 = \frac{b_2}{a_2}\bar{x}_2$$

- This course numerically solves $f(x) = 0$

Characteristic Equation

- The characteristic equation is used to study the local (linear) behavior near an equilibrium.
- The characteristic equation for a DDE is found like ODEs (Math 537), but the result is an exponential polynomial with an infinite number of solutions:

$$\begin{vmatrix} -b_1 - \lambda & f'(\bar{x}_2)e^{-\lambda R} \\ a_2 & -b_2 - \lambda \end{vmatrix} = 0$$

- This produces:

$$(\lambda + b_1)(\lambda + b_2) - a_2 f'(\bar{x}_2)e^{-\lambda R} = 0$$

- Need to find complex solutions to this equation

Characteristic Equation–Finding Eigenvalues

- The numerical simulation showed damped oscillations, which suggests that all eigenvalues have negative real part.
- The characteristic equation is studied by letting $\lambda = \mu + i\nu$, which gives

$$(\mu + i\nu + b_1)(\mu + i\nu + b_2) - a_2 f'(\bar{x}_2)e^{-\mu R}(\cos(\nu R) - i\sin(\nu R)) = 0$$

- This is solved numerically by simultaneously finding the real and imaginary parts equal to zero
- Solving two nonlinear equations in two unknowns uses vector and matrix methods to extend our technique for solving $f(x) = 0$
- We may get to these algorithms in this class, but they certainly appear in Math 693A

Characteristic Equation–Numerical Eigenvalues

- This course examines some of the basics behind the packages for solving these problems
- **MatLab** allows users to examine the coding algorithm, so knowledge from this course helps you better choose among different packages.
- We employed **Maple's** *fsolve* routine, and the first three pairs of eigenvalues with the largest imaginary parts are found:

$$\begin{aligned} \lambda_{1,2} &= -0.19423 \pm 0.98036i \\ \lambda_{3,4} &= -0.55573 \pm 3.9550i \\ \lambda_{5,6} &= -0.68084 \pm 7.07985i \end{aligned}$$

- These eigenvalues show the damped oscillatory behavior and indicate the intervals between maxima are about 2π time units.

Maple code available from Website.