Numerical Analysis and Computing

First Meeting

Lecture Notes #1

Joe Mahaffy

Department of Mathematics and Statistics

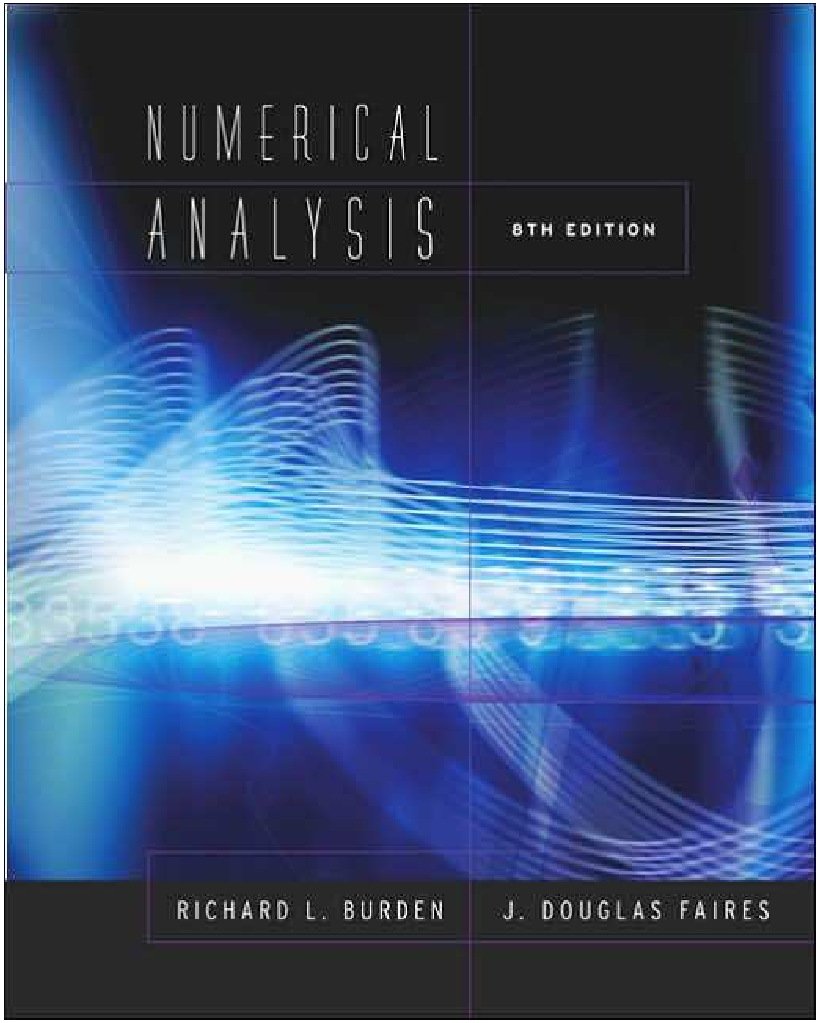
San Diego State University

San Diego, CA 92182-7720

mahaffy@math.sdsu.edu

http://www-rohan.sdsu.edu/~jmahaffy

Office	GMCS-593
Email	mahaffy@math.sdsu.edu
Web	http://www-rohan.sdsu.edu/~jmahaffy
Phone	(619)594-3743
Office Hours	M 1-2, 3:30-4:30, W 11-12, 1-2, and by appointment



Title:

"Numerical Analysis,"

8th Edition

Authors:

Richard L. Burden &

J. Douglas Faires

Publisher:

Thomson – Brooks/Cole

ISBN:

0-534-39200-8

Chapter	Title
1	Mathematical Preliminaries
2	Solutions of Equations in One Variable
3	Interpolation and Polynomial Approximation
4	Numerical Differentiation and Integration
6	Direct Methods for Solving Linear Systems
8	Approximation Theory
Math 542*: Numerical Solutions of Differential Equations*	
5	Initial-Value Problems for ODEs
11	Boundary Value Problems for ODEs
Math 543*: Numerical Matrix Analysis*	
7	Iterative Techniques in Matrix Algebra
9	Approximating Eigenvalues
Math 693a*: Advanced Numerical Analysis (Numerical Optimization)*	
10	Numerical Solution of Nonlinear Systems of Equations
Math 693b*: Advanced Numerical Analysis (Numerics for PDEs)*	
12	Numerical Solution of PDEs

Approximate grading scheme

Homework*	50%
Midterm+	25%
Final×	25%

* Both theoretical, and implementation (programming) — Matlab will be the primary programming language. However, you can program in other languages if desired, but the instructor may not be able to help.

\+ The midterm is likely to be part take-home and part in-class.

× Scheduled time: Wednesday, December 17, 13:00am – 15:00pm. (Again likely to be part take-home and part in-class.)

- Some, but not all, class attendance is OPTIONAL — Homework, projects, and announcements will be posted on the class web page.
  - REQUIRED class presentations will be posted on the class web page. (Not surprisingly, attendance is required for in-class midterms and the final exam!)
  - If you choose to attend optional classes:
    - 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.

- 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.**
- You 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!

- Missed midterm exams: Don't miss exams! The instructor reserves the right to schedule make-up exams, make such exams oral presentations, 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 WU (unauthorized withdrawal) or F will be assigned.

- **`Academic honesty`**: submit your own work — but feel free to discuss homework with other students in the class!

You need access to a computing environment in which to write your code; — you may want to use a combination of Matlab (for quick prototyping and short homework assignments) and C/C++ or Fortran (or something completely different).

There is access to computers in GMCS 422/425/428 with login **`ma_student`** and password **`FA2008`**.

You can also use the Rohan Sun Enterprise system or another capable system. [**`http://www-rohan.sdsu.edu/raccts.html`**]

Free C/C++ (**`gcc`**) and Fortran (**`f77`**) compilers are available for Linux/UNIX.

You may also want to consider buying the student version of Matlab: **`http://www.mathworks.com/`**

Math 254, or **Math 342A**

254 ⇒ **Introduction to Linear Algebra**

- Matrix Algebra, Gaussian elimination, determinants, vector spaces, linear transformations, orthogonality, eigenvalues and eigenvectors.

342A ⇒ **Methods of Applied Mathematics, I**

- Vector analysis, divergence and Stokes' theorem, integral theorems. Matrix analysis, eigenvalues and eigenvectors, diagonalization. Introduction to ODEs. Computer software for matrix applications, solving, and graphing differential equations.

CS 106, **CS 107** or **CS 205**

106 ⇒ **Intro to Programming: FORTRAN**

- Problem solving using a computer, design of algorithms.

107 ⇒ **Intro to Programming: JAVA**

- Programming methodology and problem solving. Basic concepts of computer systems, algorithm design and development, data types, program structures.

205 ⇒ **Intro to Programming and Visualization**

- Problem solving skills for science, computing/software tools of computational science, computer communications, programming and visualization.

Add Codes

Capacity:	40 students
Enrolled:	23 students
Available Add Codes:	10*

Due to ***fire regulations***, 40 students is the **hard limit**.

* See the professor after class to get add codes.

September 22 Last day to add classes, drop classes, or change grading basis. No schedule adjustments allowed after 6:00 p.m. on this date.

Questions?

- Professor Joe Mahaffy thanks Professors Peter Blomgren and Don Short for extensive access to their experience and notes for this course.
- I will borrow heavily, edit, and post on the web the notes and homework assignments created from these past instructors, especially Peter Blomgren.

- There were a few start-up problems, which should be managed soon.

- Numerical tools for problem solving:

⇒ Newton's Method for $f(x) = 0$.

⇒ Least squares approximation.

⇒ The Fast Fourier Transform (FFT).

⇒ Polynomial Interpolation.

⇒ Numerical differentiation and integration.

⇒ Taylor's Theorem.

⇒ Weierstrass' Theorem.

Q: Why are numerical methods needed?

Q: Why are numerical methods needed?

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

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?

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...

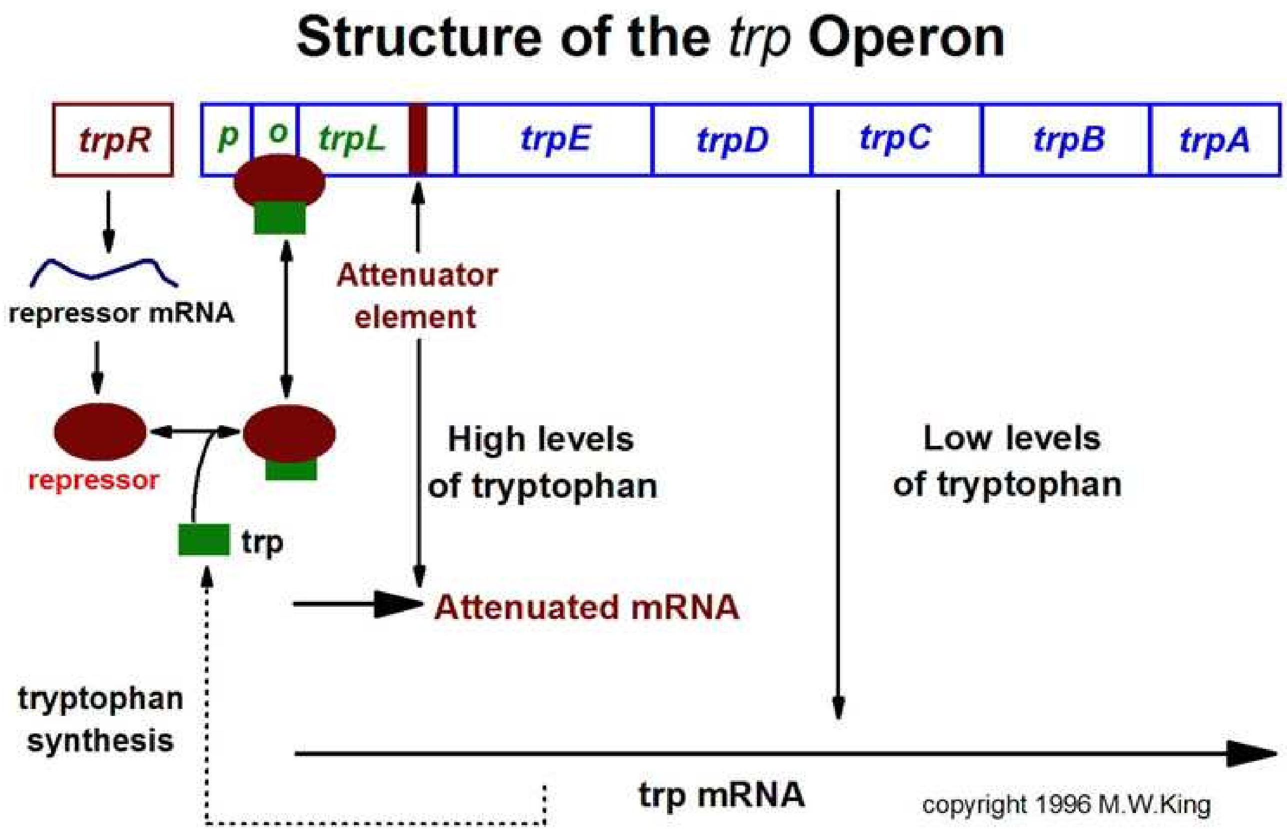
Numerical tools for problem solving:

- 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.

Genetic Control by Repression

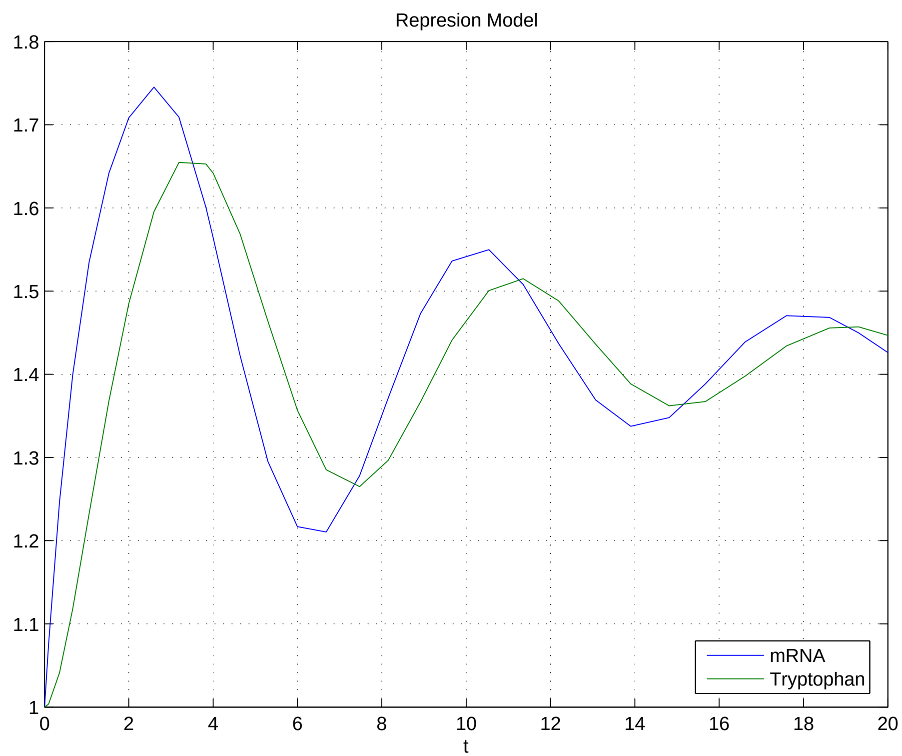


Let $x_1(t)$ be the concentration of mRNA and $x_2(t)$ be the concentration of the tryptophan (endproduct). This process is often called endproduct inhibition, and it is a negative feedback system. These systems, especially with delays, can result in oscillatory behavior.

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

This is a system of first order delay differential equations, which is infinite dimensional because of the need for initial data including a history of the solution on the interval $[-R, 0]$.

Simulated model with $a_1 = 2$, $a_2 = b_1 = b_2 = 1$, $n = 4$, and $R = 2$. MatLab simulation uses package DDE23. You will study a related algorithm in Math 542, the Runge-Kutta-Felberg method for integrating ordinary differential equations (numerically solving the ODE).



MatLab code available from Website.

- Qualitative analysis of any differential equation begins with finding all equilibria for the system.

- The equilibria are found by solving 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 gives a system of nonlinear equations equal to zero, which usually require numerically methods to approximate the equilibria. Here it easily reduces to a nonlinear scalar equation, $f(x) = 0$, which early in this course, we learn to solve.

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

- The characteristic equation is used to study the local (linear) behavior near an equilibrium.

- The characteristic equation for delay differential equations is found like one does for ordinary differential equations (Math 537), but the result is an exponential polynomial with an infinite number of solutions.

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

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

Need to find complex solutions to this equation.

- 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.

- 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 amongst different packages.

- We employed Maple's **fsolve** routine, and the first three pairs of eigenvalues with the largest imaginary parts are found.

$$\begin{aligned} \lambda_{1,2} &= -0.19423 \pm 0.98036 \\ \lambda_{3,4} &= -0.55573 \pm 3.9550 \\ \lambda_{5,6} &= -0.0.68084 \pm 7.07985 \end{aligned}$$

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

Maple code available from Website.