Math 128A Fall Quarter 2010
M.W.F. 11am - 11:50am Room 166 Chem

Instructor:	Professor A. Cheer
Office:	Room 2138 Mathematical Sciences Building
Office Hours:	Thursdays and Fridays 3pm-4:30pm
T.A.	Kristin Lui
Office:	Room 2141 MSB
Office Hours:	Tuesdays 10-11:30am and Wednesdays 3-4:40pm
Reader	TBA

Text: Numerical Analysis (Eighth Edition) by Burden and Faires. Published by Brooks/Cole.

Course Outline:

Chapter 1 (1.1-1.3)	2-3 lectures
Chapter 3 (3.1-3.6)	8-10 lectures
Chapter 4 (4.1-4.8)	7-8 lectures
Chapter 8 (8.1 - 8.6)	8-9 lectures

Grading

midterm 20 %

Either Friday Oct. 29th, 2010 or Monday Nov.1st, 2010

final exam 40%

Thursday, December 7th, 2010 10:30am-12:30pm. **No Exceptions!**

computer assignments 30%

homework assignments 10%

This is a mathematics course with a substantial programming component. You may use any programming language you like. However, it must be a general purpose programming language that does not give any special assistance in implementing the algorithms we study.

There will be five programming assignments but only the highest four scores will be counted towards your grade. All programs should be handed in on the date that they are due. Homework will be assigned throughout the quarter.

Homework assignments, computer assignments, due dates and other announcements will be posted on Smartsite. Please check it frequently.

Welcome to the class

MATH 128ABC Format for Compututation Problems

Your task in each of the programming assignments is to write a brief paper which answers the given questions and illustrates the fundamental ideas in clear, concise, descriptive English prose. The report should separate the required tasks of the given project and document each in the appropriate section, i.e. Analysis, Computer Program, or Results.

1. Analysis (30%). This section should begin by stating all the problems posed in the handout. This can be accomplished in your own words or simply by using the handout as the cover sheet to your report. Next, derivations and mathematical proofs necessary to solve the given problems or to create the required programs should be documented. A brief description of all algorithms you plan to use in your code should be given. The notation used to describe the algorithms should be consistent with the variable names and internal documentation of your programs. The last part of analysis should be a discussion of numerical considerations for the algorithms you have just described. Proofs of convergence, predicted error bounds, predicted number of iterations, etc., are examples of these considerations. They are discussed in the text or may be explained in class. If the given algorithms allow you to predict their performance before the actual computer run, then these facts and predictions should be presented.

2. Computer Program (30%). The source code should be readable and printed with margins by a printer in good working order. Internal comments should describe algorithms and variables, relating them to those described in your Analysis section. Next, this section should describe the input and output to and from your code. Some students go too far with this by showing the hand input of data to their runs- this is not necessary. A simple echoing of the data in some tabular form will suffice, as long as it is clear to the reader that you are starting with the correct initial data. Ten percent of the points for your program will be for your programming style.

3. Results (30%). This section contains the output of your program and an explanation of the results. Explaining your output should include comparing the results to the predictions (if any) in Analysis along with appropriate comparisons between the performance of different methods used and/or cases solved. If the method did not solve a given problem, explain why. If it did work, explain why these results match predicted values or are within predicted error bounds. Many students simply report in their own words the numbers from the computer output- this is not an explanation! By getting this far in the project you have covered the analysis, programmed the code, and probably run your code many times while debugging it for all the given cases. Now is the time to step back from the problem and look for patterns or comparisons in the results, and, if possible, use the information in the Analysis to explain your observations. Use your intuition and experience to make the strongest mathematically certain statements you can about the methods used applied to the given problems.

Any English course has a minimum standard for quality of written expression of ideas, and you would not consider handing in a rough draft as a final copy. The same holds true here. The projects are more than a homework assignment with a printout attached-they are a document of all of your work and understanding of the problems posed. Your report should stand on its own, documenting every aspect of the project in the context of material taught in the course. For this reason Style amounts to 10% of the total points for the project.

Incomplete problem solutions will NOT be accepted for credit.

In order to receive full credit for an assignment, it must be completed and turned in by classtime on the specified due date.

Any assignment turned in late, but on or before the absolute due date (typically the following class period) will receive a maximum of one-half credit.

Any assignment turned in after the absolute due date will not be graded and no credit will be given for it.

Start Early

MATH 128A COMPUTER PROBLEM #1 September 26th, 2010

The following three schemes can be used to recursively generate the terms in the sequence $\left[(\frac{1}{3})^n\right]_{n=0}^{\infty}$.

$$r_0 = 1 \quad \text{and} \quad r_n = \frac{1}{3}r_{n-1} \quad \text{for n} = 1, 2, \cdots, \tag{1}$$

$$p_0 = 1, \quad p_1 = \frac{1}{3} \quad \text{and} \quad p_n = \frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2} \quad \text{for n} = 2, 3, \cdots, \tag{2}$$

$$q_0 = 1, \quad q_1 = \frac{1}{3} \quad \text{and} \quad q_n = \frac{10}{3}q_{n-1} - q_{n-2} \quad \text{for n} = 2, 3, \cdots, \tag{3}$$

(A) By direct substitution (i.e. analytically) verify that the general solution to (2) is given by

$$p_n = A(\frac{1}{3})^n + B.$$

Setting $A = 1$ and $B = 0$ will generate the desired sequence.

(B) Verify that the general solution for equation (3) is given

$$q_n = A(\frac{1}{3})^n + B3^n.$$

Setting $A = 1$ and $B = 0$ will generate the desired sequence.

(C) Write a well documented computer program to generate approximations to the sequence $x_n = (\frac{1}{3})^n$ using the three schemes [equations (1), (2) and (3) above]. The initial error is in the 5th digit as given below.

$$r_0 = 0.99996 \quad \text{and} \quad r_n = \frac{1}{3}r_{n-1} \quad \text{for n} = 1, 2, \cdots, \tag{1a}$$

$$p_0 = 1, \quad p_1 = .33332 \quad \text{and} \quad p_n = \frac{4}{3}p_{n-1} - \frac{1}{3}p_{n-2} \quad \text{for n} = 2, 3, \cdots, \tag{2a}$$

$$q_0 = 1, \quad q_1 = .33332 \quad \text{and} \quad q_n = \frac{10}{3}q_{n-1} - q_{n-2} \quad \text{for n} = 2, 3, \cdots, \tag{3a}$$

(D) Discuss the results obtained and investigate (analytically) the propagation of errors for each scheme.

DUE DATE: *Friday, October* 8*th*, 2010 (*Classtime*)

Absolute Deadline: *Monday, October* 11*th*, 2010 (*Classtime*)

In order not to loose points on the write up of your project, please refer to the handout **Format** for Computer Problems posted. This is going to be the easiest program of the quarter, so don't choose this one to skip.

math 128A
numerical analysis · MW 11–11^50. Sept 27, 2010

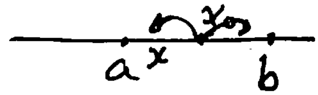
- last friday was review of Calculus.
- for Taylor series, need the function to be ∞ differentiable.

if $n+1$ differentiable only, then will only get n degree polynomial.

Suppose $f \in C^n[a,b]$ and $f^{(n+1)}$ exists on $[a,b]$

let $x_0 \in [a,b]$, for every $x \in [a,b]$ $\exists\ \xi(x)$

between x_0 and x with

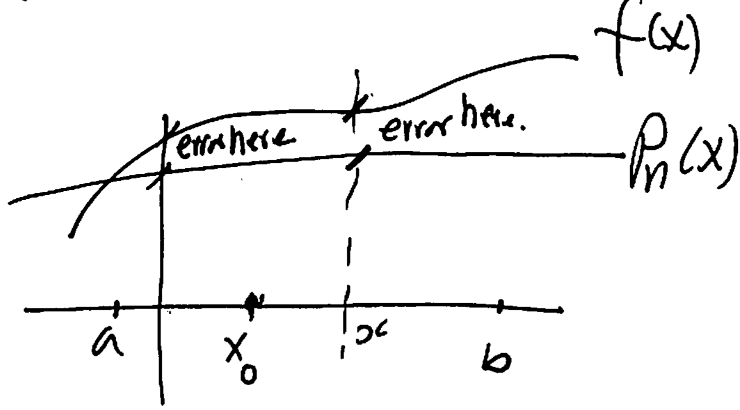


$$f(x) = P_n(x) + R_n(x)$$

with $P_n(x) = f(x_0) + f'(x_0)(x-x_0) + \cdots$

$$R_n(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x-x_0)^{n+1}$$

→ number that depends on x ?



this series expansion is only good around x_0.

Numerical analysis 8th edition.

example

$f(x) = \cos(x)$

expand around $x = 0$

so $\cos(x) = 1 - \frac{x^2}{2}$ ~~$+ \frac{1}{24}x^4 + \cos$~~

$$+ \underbrace{\frac{1}{24} x^4 \cos(\xi)}_{R_n(x)}$$

how far in series do you need to go to guarantee that the difference between $P_n(x)$ and $f(x) \leq$ some value?

so need to find an n so that the remainder is less than that value.

on computer, can't represent things exactly.

suppose want to represent on 32 bit.

$$.00000000123456789$$

① $0.1234567 \times 10^{-14}$ truncation

② $0.1234568 \times 10^{-14}$ rounding.

if we do $1 +$ (.00000000123456789) we get 1. i.e. lost all the digits.

So when adding 2 numbers one too small, and one too large, then we get a problem.

assume $|f(x) - P(x)| < 0.000001$

If we multiply this by Large # M, then we lose digits.

So multiplying Large # by small #.

Same as dividing Large # by small # (?).

how to measure these error?

if p^* is an approximation to p, then

absolute error is $|p - p^*|$ and

relative error is $\dfrac{|p - p^*|}{|p|}$ $\quad p \neq 0.$

p^* is said to approximate p to t significant digits if t is the largest non-negative integer for which $\dfrac{|p - p^*|}{|p|} < 5 \times 10^{-t}.$

---

example $f(x) = x^3 - 6x^2 + 3x - 0.149.$

to evaluate $f(x)$ at $x = 0.5$

7 multiplies
3 add/subtract.

each time this is done, error accumulates.

example, $f(4.71)$ gives relative err $\simeq 0.04$
3 digit chopping.

but we can write

$$= x^3 - 6x^2 + 3x - 0.149$$

$$= x(x^2 - 6x + 3) - 0.149$$

$$= x(x(x-6)+3) - 0.149.$$

so now have 2 multiply, 3 add/sub. better.
this is called Horner.
relative error with 3 digit rounding is 0.0045

so we get one more digit of accuracy.
so gain one order of mag.

Homework Assignment #1

Due Friday October 1st, 2010

Section 1.1	#3ab, 4d, 6, 9a, 12, 25, 26
Section 1.2	#4d, 10b, 12, 20a, 24, 26
Section 1.3	#6, 9

Please feel free to do as many problems as you like for fun.