Numerical Analysis and Computing
Lecture Notes #1 — First Meeting

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720
http://terminus.sdsu.edu/

Fall 2014

Academic Life
MSc

- MSc. Engineering Physics, Royal Institute of Technology (KTH), Stockholm, Sweden. Thesis Advisers: Michael Benedicks, Department of Mathematics KTH, and Erik Aurell, Stockholm University, Department of Mathematics. Thesis Topic: “A Renormalization Technique for Families with Flat Maxima.”

Figure: Bifurcation diagram for the family $f_{a,1/2}$ [Blomgren-1994]

PhD


Figure: The noisy (SNR = 4.62 dB), and recovered space curves. Notice how the edges are recovered. [Blomgren-1998]
Research Associate. Stanford University, Department of Mathematics. Main Focus: Time Reversal and Imaging in Random Media (with George Papanicolaou, et. al.)

Figure: Comparison of the theoretical formula for a medium with $L = 600 \, m$, $a_e = 195 \, m$, $\gamma = 2.12 \times 10^{-5} \, m^{-1}$. [LEFT] shows a homogeneous medium, $\gamma = 0$, with $a = 40 \, m$ TRM (in red / wide Fresnel zone), and a random medium with $\gamma = 2.12 \times 10^{-5}$ (in blue). [RIGHT] shows $\gamma = 0$, with $a = a_e = 195 \, m$ (in red), and $\gamma = 2.12 \times 10^{-5}$, with $a = 40 \, m$ (in blue). The match confirms the validity of the theory. [BLOMGREN-PAPANICOLAOU-ZHAO-2002]

Professor, San Diego State University, Department of Mathematics and Statistics. Projects: Computational Combustion, Biomedical Imaging (Mitochondrial Structures, Heartcell Contractility, Skin/Prostate Cancer Classification).

Figure: [LEFT] Phase-space projections produced by the time coefficients of the POD decomposition of the rotating pattern shown in [RIGHT]. [BLOMGREN-GASNER-PALACIOS-2005]

Fun Times... ⇒ Endurance Sports

- Triathlons:
  - (10) Ironman distance $(2.4 + 112 + 26.2) \rightarrow 11:48:57$
  - (15) Half Ironman distance → 5:14:20

- Running
  - (1) Trail Double-marathon (52 miles) → 10:59:00
  - (4) Trail 50-mile races → 9:08:46
  - (6) Trail 50k (31 mile) races → 5:20:57
  - (12) Road Marathons → 3:26:19 (7:52/mi)
  - (18) Road/Trail Half Marathons → 1:36:25 (7:21/mi)
Basic Information: The Book

Class Conceptualized out of 7th (5th?) Edition

Title: “Numerical Analysis,”
9th Edition (older OK)

Authors: Richard L. Burden & J. Douglas Faires

Publisher: Brooks/Cole CENGAGE Learning


Basic Information: Syllabus

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mathematical Preliminaries</td>
</tr>
<tr>
<td>2</td>
<td>Solutions of Equations in One Variable</td>
</tr>
<tr>
<td>3</td>
<td>Interpolation and Polynomial Approximation</td>
</tr>
<tr>
<td>4</td>
<td>Numerical Differentiation and Integration</td>
</tr>
<tr>
<td>5</td>
<td>Direct Methods for Solving Linear Systems</td>
</tr>
<tr>
<td>6</td>
<td>Approximation Theory</td>
</tr>
<tr>
<td>7</td>
<td>Numerical Solutions of Differential Equations</td>
</tr>
<tr>
<td>8</td>
<td>Initial-Value Problems for ODEs</td>
</tr>
<tr>
<td>9</td>
<td>Boundary Value Problems for ODEs</td>
</tr>
<tr>
<td>10</td>
<td>Iterative Techniques in Matrix Algebra</td>
</tr>
<tr>
<td>11</td>
<td>Approximating Eigenvalues</td>
</tr>
<tr>
<td>12</td>
<td>Numerical Solution of Nonlinear Systems of Equations</td>
</tr>
<tr>
<td>13</td>
<td>Numerical Solution of PDEs</td>
</tr>
<tr>
<td>14</td>
<td>Advanced Numerical Analysis (Numerical Optimization)</td>
</tr>
<tr>
<td>15</td>
<td>Advanced Numerical Analysis (Numerics for PDEs)</td>
</tr>
</tbody>
</table>

Recent (and Growing) Influence on the Class

Title: “Approximation Theory and Approximation Practice,”

Author: Lloyd N. Trefethen

Publisher: SIAM


Basic Information: Grading

<table>
<thead>
<tr>
<th>Homework*</th>
<th>25%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm #1+</td>
<td>25%</td>
</tr>
<tr>
<td>Midterm #2+</td>
<td>25%</td>
</tr>
<tr>
<td>Final×</td>
<td>25%</td>
</tr>
</tbody>
</table>

* There will be ≈8 homework assignments; they will be posted and submitted thru http://webwork.sdsu.edu/
+ The midterms(s) may be take-home (target date(s): week #6, week #11.)
× Scheduled time: Tuesday, Dec 16, 1:00pm – 3:00pm. (Details to be determined.)
Homework ~ Webwork

“WeBWorK is a web-based interactive system designed to make homework in mathematics and the sciences more effective and efficient.”

- Info resource
  http://webwork.maa.org/wiki/Category:Students
- Homework will "open" (on http://webwork.sdsu.edu/), on the first day material relevant to the HW is covered in class.
- Homeworks will "close" (be due), no less than 8 days after the last material relevant to the HW is covered in class.
- Your Lastname is your LoginID, and RedID is your initial password.
  - If your Lastname is not unique, then your loginID is Lastname+Firstname-initial.

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.

Homework ~ Webwork

- Most HW problems involve both a theoretical, and implementation (programming) part — Matlab is the recommended and supported environment, but feel free to program in 6510 assembler, Java, Fortran, C/C++, M$-D♭... You will enter specific results from your code into Webwork.
- Some assignments may require additional hardcopy submissions; read the workwork instructions.
- Start Assignments EARLY.
- Sometimes “interesting” things happen in Webwork (which does computations using Perl), which make Webwork computations different from Matlab computations.

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 WU or F will be assigned.

- Academic honesty: submit your own work — but feel free to discuss homework with other students in the class! It’s OK to ask “Uncle Google” and “Aunt Wiki” for help and ideas, but process the information and make it your own, AND cite any and all sources you use.

Honesty Pledges, I

- The following Honesty Pledge must be included in all programs you submit (as part of homework and/or projects):
  - I, (your name), pledge that this program is completely my own work, and that I did not take, borrow or steal code from any other person, and that I did not allow any other person to use, have, borrow or steal portions of my code. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of the San Diego State University Policies.
  - Work missing the honesty pledge may not be graded!

Honesty Pledges, II

- Larger reports must contain the following text:
  - I, (your name), pledge that this report is completely my own work, and that I did not take, borrow or steal any portions from any other person. Any and all references I used are clearly cited in the text. I understand that if I violate this honesty pledge, I am subject to disciplinary action pursuant to the appropriate sections of the San Diego State University Policies. Your signature.
  - Work missing the honesty pledge may not be graded!

Computer Resources

- Access to a (somewhat) current release of Matlab is highly recommended.

- The GMCS-422/428 labs will be available.

- You can also use the Rohan Sun Enterprise system or another capable system.

- How to open a ROHAN account: [http://www-rohan.sdsu.edu/raccts.shtml](http://www-rohan.sdsu.edu/raccts.shtml)

- You may also want to consider buying the student version of Matlab: [http://www.mathworks.com/](http://www.mathworks.com/)

- SDSU students can download a copy of matlab from [http://www-rohan.sdsu.edu/~download/matlab.html](http://www-rohan.sdsu.edu/~download/matlab.html)
Math 541: Formal Prerequisites

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**

---

Math 541: Introduction — What we will learn

- Numerical tools for problem solving:
  - $\Rightarrow$ **powertool** Newton’s Method for $f(x) = 0$.
  - $\Rightarrow$ **powertool** Least squares approximation.
  - $\Rightarrow$ **powertool** The Fast Fourier Transform (FFT).
  - $\Rightarrow$ **tool** Polynomial Interpolation.
  - $\Rightarrow$ **tool** Numerical differentiation and integration.

- Foundation:
  - $\Rightarrow$ **foundation** Taylor’s Theorem.
  - $\Rightarrow$ **foundation** Weierstrass' Theorem.

---

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...

---

Peter Blomgren, ⟨blomgren.peter@gmail.com⟩ Lecture Notes #1 — First Meeting — (21/33)
Computing: Historical Perspective...

The Apollo Guidance Computer (1969)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word Length</td>
<td>15 bits plus parity</td>
</tr>
<tr>
<td>Fixed Memory Registers</td>
<td>36,864 Words</td>
</tr>
<tr>
<td>Erasable Memory Registers</td>
<td>2,048 Words</td>
</tr>
<tr>
<td>Number of Normal Instructions</td>
<td>34</td>
</tr>
<tr>
<td>Number of Involuntary Instructions</td>
<td>10</td>
</tr>
<tr>
<td>Number of Interface Circuits</td>
<td>227</td>
</tr>
<tr>
<td>Memory Cycle time</td>
<td>11.7 µs (85 kHz)</td>
</tr>
<tr>
<td>Addition Time</td>
<td>23.4 µs</td>
</tr>
<tr>
<td>Multiplication Time</td>
<td>46.8 µs</td>
</tr>
<tr>
<td>Number of Logic Gates</td>
<td>5,600 (2,800 packages)</td>
</tr>
<tr>
<td>Volume</td>
<td>0.97 cubic feet</td>
</tr>
<tr>
<td>Weight</td>
<td>70 pounds</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>55 watts</td>
</tr>
</tbody>
</table>

Counting Work: The Memory Access Latency Model

If we have three cache-levels (L1, L2, and L3), some average hit-rate (and hence miss-rate) for each level and the time it takes to access that cache-level (the hit-cycle-time), then we end up with a measure for the average memory access latency per memory access

\[ T \sim \left( L1 \text{hit rate} \times L1 \text{hit cycle time} \right) \\
+ \left( L1 \text{miss L2 hit rate} \times L2 \text{hit cycle time} \right) \\
+ \left( L2 \text{miss L3 hit rate} \times L3 \text{hit cycle time} \right) \\
+ \left( L3 \text{miss rate} \times \text{[S]DRAM latency} \right) \]

If this does not scare you, please get a Ph.D. in algorithm design on the compiler / silicon level!!!

Meanwhile, the rest of us will count “flops”, i.e. floating-point operations (multiplications and additions)!

Counting Work: The Memory Access Latency Model

Performance Impact of Accessing Memory (Matrix)

We need some measure of how fast, or slow, an algorithm is... (independent of the computational architecture)

In the old-old days multiplications (and divisions) where a lot slower additions (and subtractions) \( T_{*,/} \gg T_{+,-} \); so one would count the number of multiplications (see e.g. the timings for the Apollo guidance computer.)

Then chip designers figured out how to make multiplications faster, so \( T_{*,/} \approx T_{+,-} \), so in the old days one would count all operations.

Yesterday, processors where so fast that memory accesses dominated the processing time; in particular cache-misses, so we end up with a completely different model...

Matlab Code

```matlab
for pow=1:14
    n=2^pow;
    clear A
    A = zeros(n,n);
    a0 = clock;
    for c=1:n
        for r=1:n
            A(r,c) = 1/(1+r+c);
        end
    end
    a1 = clock;

    clear B
    B = zeros(n,n);
    b0 = clock;
    for c=1:n
        for r=1:n
            B(r,c) = 1/(1+r+c);
        end
    end
    b1 = clock;

    a_time = [a_time etime(a1,a0)];
    b_time = [b_time etime(b1,b0)];
end
plot((1:14),b_time./a_time,'o-')
ax = axis;
axis([0 15 ax(3:4)])
grid on
xlabel('n, (Matrix Size = 2^n \times 2^n)')
ylabel('Timing ratio')
title('Row-Column vs. Column-Row Matrix Fill')
```
Our Universe...

Clearly, the question **“How fast is [any given] algorithm?”** has many answers.

For starters, we will keep things simple, and think about old “flops” measurements.

However, it is good to be aware that other aspects of the computer architecture have great impact on the overall performance.

Note that the results in the previous example(s) are mathematically equivalent, but in the vector-fill example the worst/best-timing-ratio for the vector of size $2^{17}$ is $\sim 10^4$. 

---

**Code**

```matlab
matlab_time = []; prefill_time = []; loop_time ==[];
for pow=1:17
    n=2^pow;
    \% Matlab
clear test_vec
test_vec=zeros(1,n);
b0 = clock;
for k=1:n
test_vec(k) = k;
end
b1 = clock;
matlab_time = [matlab_time etime(a1,a0)];
prefill_time = [prefill_time etime(b1,b0)];
loop_time = [loop_time etime(c1,c0)];
end
semilogy(1:max(pow),matlab_time,'b-', 1:max(pow),prefill_time,'ro',
1:max(pow),loop_time,'ks')
legend('Matlab','Prefilled','Unfilled','Location','NorthWest')
xlabel('n, where vector-length is $2^n$')
ylabel('Timing')
title('Vector-fill Timing')
grid on
ax = axis;
axis([0 max(pow+1 ax(3:4)])
```
The Professor
The Class — Overview
The Class...
Introduction
The What? Why? and How?
Perspective
Performance Measures...
References

Links

- http://terminus.sdsu.edu/SDSU/Math541_f2014/
  - The Class website
- http://webwork.sdsu.edu/
  - Webwork server, for homework.
- http://webwork.maa.org/wiki/Category:Students
  - Some information about Webwork
- http://www-rohan.sdsu.edu/raccts.shtml
  - "Obtaining a ROHAN Computer Account"
- http://www.mathworks.com/
  - Mathworks, the makers of Matlab
- http://www-rohan.sdsu.edu/~download/matlab.html
  - On-campus matlab download.