My favourite books, papers and scientific software

Denys Dutykh

CNRS INSMI – LAMA UMR 5127
Université Savoie Mont Blanc
73376 Le Bourget-du-Lac France

July 17, 2015

Abstract

In this document I decided to assemble various sources of information (books, papers) and also some scientific software libraries which turned out to be extremely useful in the scientific work of the author of these notes. Namely, my vision of some fields of (Applied) Mathematics and Fluid Mechanics has been strongly influenced by these manuscripts. In the past I read far more books and scientific papers. However, only those listed below had the most influential effect on me. As a conclusion, I can strongly recommend to consult these references and absorb the precious fundamental knowledge from them.

The reader will notice that most of my favourite references are rather old. It can be mainly explained by the personal taste of the author of these notes and an additional purely subjective observation: it seems that in older times the quality and information content of the books were much higher...

Contents

Chapter 1

Books

In this part of the document I shall describe the books which influenced and shaped somehow my vision of (Applied) Mathematics and Fluid Mechanics. The majority of these books are written in English. However, some are in Russian or in French. However, in most cases it is possible to find their English translations.

1.1. Calculus

To my taste these two volumes constitute probably the best and most complete introduction into the real analysis. It incorporates already the first notions of General Topology, but the exposition is not at all abstract, as one could think. The proofs are clear and complete. In general, the theory is illustrated with many examples of calculations. For example, the first volume is finished by Morse Lemma and the theory of conditional extremum. The second volume contains some information on differential forms and their integration theory. To read absolutely if you want to revise the real analysis and not only....

- Zorich, V. A. (2008). *Mathematical Analysis I* (2nd ed.). 574pp. Berlin Heidelberg New York: Springer Verlag.
- Zorich, V. A. (2008). *Mathematical Analysis II* (2nd ed.). 688pp. Berlin Heidelberg New York: Springer Verlag.

1.2. Advanced calculus

This book is an excellent introduction into the exterior calculus, differential forms and integration theory on manifolds (up to the general form of Stokes theorem). Perhaps, it is the best book to read in the first place to be introduced into the Calculus on manifolds:

- Spivak, M. (1971). *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*. Princeton: Westview Press.

1.3. Complex analysis

The following book is probably the best course of (advanced) Calculus ever made so far. First of all, there is no separation between the real and complex analysis. Both disciplines live together and enrich each other. The Riemann surfaces are implicitly introduced already in the sixth lesson... The famous interpolation scheme of Hermite can be found there as well with an elegant original proof. Numerous tricks to compute exactly some definite (proper and improper) integrals are also explained in this book along with the study of some special functions. Despite the age of this book, it can be read without any difficulties even today. All notations are modern.

- Hermite, C. (1891). *Cours de M. Hermite rédigé en 1882 par M. Andoyer, élève à l'École Normale* (Quatrième édition, revue et augmentée). Paris: Librairie Scientifique A. Hermannn. 8, rue de la Sorbonne.

1.4. Differential equations

1.4.1 Ordinary Differential Equations

A fairly good and rather short introduction into the theory of ODEs. During many years I. G. Petrovsky was the president of the Moscow State University. His colleagues were A. N. Kolmogorov, I. M. Gelfand and many other brilliant mathematicians. My main motivation to read this book was the desire to take a snapshot of the ODE theory state before V. I. Arnold (Kolmogorov & Moser).

- Petrovsky, I. G. (1984). *Lectures on the theory of ordinary differential equations*. (A. D. Myshkis & O. A. Oleinik, Eds.). Moscow: Moscow University Press (in Russian).

1.4.2 Partial Differential Equations

1.5. Numerical analysis

1.6. Applied Mathematics

This tiny book was published in 2000, but it is an expanded version of a talk delivered by V.I. Arnold in 1997. This text of Arnold is truly accessible even

for a general public and it contains some deep thoughts on the mathematical modelling and applied mathematics in general. These considerations are illustrated with simple examples from various applied fields ranging from Biology to Statistics and Probabilities (I do not know whether it was translated into English...):

- Arnold, V.I. (2000) *"Rigid" and "soft" mathematical models*. 32pp. Moscow: MCCME (in Russian)

1.7. Fluid mechanics

Perhaps the best introductory text into Fluid Mechanics is the following book:

- Chorin, A., & Marsden, J. (1993). *A Mathematical Introduction to Fluid Mechanics* (3rd ed.). Springer.

My personal lecture notes on Hydrodynamics are based in part on the book mentioned above. If you want to have a look onto my own piece of work, you can download it for free at this URL:

- https://github.com/dutykh/hydro/

1.8. Geometry

1.8.1 Differential Geometry

1.9. Topology

This elementary book on Topology appeared first in German in 1932. P. S. Aleksandrov was a close friend of D. Hilbert and he worked on this manuscript during his stays in Germany (Göttingen). Only later it was translated into Russian and published in the Soviet Union. The particularity of this book is that it describes on an elementary level the state of (a new field by that time) Topology after H. Poincaré and before N. Bourbaki:

- Aleksandrov, P. S., & Efremovich, V. A. (1936). *Outline of the main notions of topology*. Moscow: ONTI — NKTP USSR (in Russian).

Chapter 2

Papers

2.1. Floating point arithmetics

The following reference contains several striking examples on how bad rounding errors can be. Moreover, some drawbacks in floating-point arithmetics realizations in popular software (*e.g.* Microsoft Excel) are highlighted. In my opinion, this text has to be read by any practitioner of numerical computations:

- Kahan, W. (2006). *How futile are mindless assessments of roundoff in floating-point computation?* Work in progress, 56 pp.

In some situations the peculiarities of floating-point computations can lead to totally wrong and/or unpredictable results. As a general recommendation to diagnose such eventual floating-point arithmetics problems, Prof. W. Kahan [Kah06, §4] suggests to follow this scheme:

- Repeat the computation in arithmetics of increasing precision. Significant digits in the numerical result must agree.
- Repeat the computation in arithmetics of the same precision but rounded differently (towards zero or towards infinity) and compare these results.
- Repeat the computation in arithmetics of the same precision but rounded randomly and treat the results statistically.
- Repeat the computation in arithmetics of the same precision but with slightly perturbed input data and see how the results spread.
- Perform the computations in *interval arithmetics* to obtain a bound on results.

However, one has to remember that the effects of round-off cannot be in general assessed without a mathematically rigorous and time-consuming error analysis [Kah06].

2.2. Software engineering

And to make a smooth transition to the next part of the document, we discuss some software engineering principles, which are useful in scientific programming.

- Kelly, D., Hook, D. & Sanders R. *Five Recommended Practices for Computational Scientists Who Write Software.* Comp. Sci. Eng., **11**(5), 48–53, 2009
- Wilson, G. *et al. Best Practices for Scientific Computing.* PLOS Biology, 2014

This article discusses some aspects of scientific aspects, but its main strength lies in the incitation to reproducible research and Open science in general:

- LeVeque, R. J., Mitchell, I. M. & Stodden, V. *Reproducible research for scientific computing: Tools and strategies for changing the culture.* Comp. Sci. Eng., **14**(4), 13–17, 2012

Chapter 3

Software libraries

As a programming language I usually prefer to use C/C++. I can work with Fortran codes, but it is not the choice I would make to develop new software today. This vision explains some choices given below.

3.1. Integrated Development Environments

Taking into account the remark given just above, here are my two favourite IDEs:

- CLion: https://www.jetbrains.com/clion/
- Eclipse: https://eclipse.org/ide/

3.2. Compilers

This Section will be rather short since in my work I use only one family of compilers – the GNU Compiler Collection (GCC). It is good and free. Perhaps, its performance can be beaten by Intel Compilers in some cases, but it requires a lot of fine tuning. So, my choice is the freedom and portability of GCC:

- https://gcc.gnu.org/

3.3. Linear algebra

The codes I develop are mostly intended to be run on my laptop. That is why I usually privilege lightweight libraries. It is not the best choice for codes which will be run on massively parallel computers. For such users I could recommend using exclusively PETSc library. However, for more modest users like the Author of this document, the following choices can be considered:

- Eigen: http://eigen.tuxfamily.org/

- Armadillo: http://arma.sourceforge.net/
- Blaze-lib: https://code.google.com/p/blaze-lib/

3.4. Finite Element Libraries

This is a generic open source finite element library which supports the adaptive mesh refinement. An important restriction: this library deals only with quadrangle elements:

- https://www.dealii.org/

This is my favourite finite element library under Matlab environment:

- https://code.google.com/p/felicity-finite-element-toolbox/

3.5. Graphics

`PLplot` is a cross-platform software package for creating scientific plots in compiled languages like C/C++, Fortran, *etc.* :

- http://plplot.sourceforge.net/

3.6. Fourier transforms

This library is the *de facto* standard in FFT computation:

- http://www.fftw.org/

Contrary to the previous library, this tool is a lightweight library to do the same job:

- http://sourceforge.net/projects/kissfft/

And finally, a library which allows to perform the Fourier analysis on non-uniform grids:

- http://www.cims.nyu.edu/cmcl/nufft/nufft.html

UPDATE: Recently I discovered a new development in this domain which looks very promising: Sparse Fast Fourier Transform. This algorithm uses the sparsity of the Discrete Fourier Transform (DFT) to accelerate the computations over FFT. The advantage of this approach for numerical analysis applications is yet to be confirmed, however.

- http://groups.csail.mit.edu/netmit/sFFT/

Chapter 4

Software

4.1. Computing

It is not probably the right place for this software, but I have to speak about it. Yes, in my work I use quite a lot of MATLAB. It allows you to implement really fast numerical algorithms and to visualize the results immediately to produce publication quality graphics. And for those who say that Matlab is slow – it is not true anymore. Engineers of Mathworks did a tremendous work in optimizing and performance tuning of Matlab to the point that nowadays the backslash operation \ and FFT functions `fft()` are practically unbeatable.

- http://www.mathworks.com/products/matlab/

4.1.1 Matlab tools

Advanpix toolbox equips MATLAB with a new multiple precision floating-point numeric type and extensive set of mathematical functions that are capable of computing with arbitrary precision. It is probably the easiest and the most efficient way to perform multiprecision computations today:

- http://www.advanpix.com/

Exporting a figure from MATLAB the way you want it, can be a real headache for the unitiated. The next toolbox `export_fig` was designed for exporting figures from MATLAB to standard image and document formats (*e.g.* jpg, eps, pdf, *etc.*) with high publication-ready quality:

- https://github.com/altmany/export_fig/

4.2. Symbolic computations

Concerning the symbolic computations I shall not be very original. As many other users throughout the world, I use equally well Maple & Mathematica (however, with slight preference given to Maple):

- Maple: http://www.maplesoft.com/products/maple/
- Mathematica: https://www.wolfram.com/mathematica/

4.3. Graphics

In order to make the plots/sketches to illustrate some ideas I usually use two softwares. For relatively simple plots I employ LaTeXDraw which allows to export vector drawings in PSTricsk code:

- http://latexdraw.sourceforge.net/

For more complex drawings I use Inkscape:

- https://inkscape.org/

4.4. Visualization

Two classical and probably best choices are:

- Paraview: http://www.paraview.org/
- VisIt: https://wci.llnl.gov/simulation/computer-codes/visit/

Bibliography

[Kah06] W. Kahan. How futile are mindless assessments of roundoff in floating-point computation? Technical report, 2006.

Index