Study guide for MMA691 Project in Mathematics Fall 2016

Course homepage:

On the course homepage you can find the study guide and relevant links. Project descriptions will also be posted there.
`www.mdh.se/amnen/matematik/kurser/kurshemsidor/mma291/mma691-project-in-mathematics-1.45674`

Objectives:

The course shall:

- Give a deeper knowledge of a mathematical subject
- Develop the ability to formulate a problem and to work independently
- Develop the ability to present knowledge and the results obtained in speech and writing.

Examination:

PRO1, 7.5 credits, marks Pass (G) or Pass with distinction (VG), Project with oral presentation.

The project will be done individually or in pairs (if possible) and each person (pair) will choose a project topic and write, in cooperation with the supervisor, a description of their project which describes the chosen topic and the goals of the project. The description should include a descriptive headline for the project (not necessarily final), a description of the topic with some concrete goals and at least one source that can be used for the project. An example of a project description can be found on the course homepage.
The focus of the project should be mathematical but can be mathematical in different ways. It can consist of

- *mathematical modelling*, that is looking at ways to interpret theories mathematically to build mathematical models or comparing different mathematical models,
- *mathematical problem solving*, where a specific problem and ways of solving it are described in detail are examined or
- *mathematical computation*, where a model or mathematical technique is used to analyse a set of data, focusing on the mathematical method / model, it's advantages, disadvantages and implementation.

Naturally these different areas can also be combined.

Literature:

The literature depends on the chosen topic. Each group or students supervisor may supply the students with some appropriate literature but searching for relevant information on your own is part of the course.

Instruction:

After choosing topic the student is assigned a supervisor. Continuous supervision of the project is mandatory. The student and supervisor should agree on how this supervision should be handled.

Oral presentation

The oral presentation of the project will be 30 minutes per person including some time for discussion and answering questions. The date for it will be decided when a sufficient amount of progress has been made on the project.

Written report

Sources and citing

All sources for the project must be clearly written in the report. If text is copied from any source then it should be given as a quotation and be appropriately attributed. The exception to this is definitions and theorems. Definitions of common and well-known concepts can be given without a source, as can well-known 'standard' theorems. Less well-known definitions and theorems should be attributed but should not be given as quotes. For examples of one way to handle citations see the topic suggestions.
Wikipedia should not be used as a source! It is often a good idea to look at the sources given in the wikipedia article in order to find good sources though. The same goes for most encyclopaedias and reference works.

Report formatting

The length of the report depends on the project and is decided upon together with the supervisor. The report should be structured clearly and have a table of contents. All tables and figures should be numbered and given a descriptive caption. Important equations and formulas should be numbered.

LaTeX

The report should be computer written using the LaTeX typesetting software. LaTeX is a free and open source software available for most platforms that is designed to make it easy to write reports, articles and books of a technical nature. The program can be tricky to learn but many journals an conferences require submissions to be written using LaTeX and the program is very powerful in the hands of an experienced user.
There will be workshops during the semester where the use of LaTeXwill be demonstrated. More information will be available on the Blackboard page for the course.

Topic suggestions

Methods for solving least-square problems

In many instances when analysing real data it can be useful to perform least squares regression with respect to some particular type of model. Suppose we have a model that is function (uni- or multivariate) described by a set of parameters, $f(\vec{x};\vec{p})$, then a least-square problem is the problem of finding the parameter values, $\vec{p}$, that minimises the sum of the squares of the residuals for some set of data, y_k, $k = 1, \dots, n$, in other words, find $\vec{p}$ such that $\sum_{k=1}^{n}(y_k - f(\vec{x}_k;\vec{p}))^2$ is minimized.

Most likely you have already encountered this for certain types of model functions, primarily polynomials. The purpose of this project is to learn more about other kinds of least-squares problem and some methods related to solving them. For the project you should choose at least one type of generalized least-square problem and write a report about it, some suggestions are:

- Constrained least-squares problems are a common version of least squares problems where some further conditions has been imposed on the parameters. A common example is that we force all parameters to be non-negative. See for example chapter 21-23 in [1] or chapter 5 in [2].
- Non-linear least-square problems. There are a number of method that can be used to solve a least square-problem where $f(\vec{x};\vec{p})$ does not depend linearly on $\vec{p}$. Two well-known examples are the Marquardt-Levenburg method, see [3], or trust-region methods, see chapter 9 in [2].

You can also decide to implement some more sophisticated method using suitable programming software. Both [1] and [2] contains several alternative methods for solving both linear, constrined and non-linear least-square problems. Another useful source might be [4].
For suggestions of specific least-square problems to study and data to test your algorithms on, contact your supervisor.

The Korteweg–de Vries equation and shallow water waves

The Korteweg–de Vries (KdV) equation is a partial differential equation that has a long history of being used to model waves in shallow water (or other liguids or more exotic situations such as magnetohydrodynamic waves in plasma, acoustic waves in an anharmonic crystal and the 'Great Red Spot' on Jupiter, see [5] and the references therein), especially of the type called *solitons*. The equation is usually written

$$\frac{\partial u}{\partial t} + u(x,t)\frac{\partial u}{\partial x} + \delta^2 \frac{\partial^3 u}{\partial x^3} = 0$$

and is a classical example of a nonlinear partial differential equation, partially due to the fact that it can be solved exactly. Even though exact solutions can be constructed the KdV equation is usually studied using numerical schemes, especially when examining how the solutions behave over time with different initial and boundary conditions. In this project you should familiarize yourself with some of the basic properties of the KvD-equation. This can be done in different ways.
In this project you should

- Describe the typical KdV equation and its cnoidal wave solution and right-moving solitary wave solution. There are many places to find this kind of basic description, see for example [7, 5].

and also do at least one of the following

- Describe where the KdV equation arises and some of its variations, there are many modifications of the KdV equation and other equations that approximate the behaviour of shallow water waves. An alternative approach is to derive the KdV equation in some more complicated setting, such as plasma physics. To find inspiration for this part, you can start at the Wikipedia page for the Korteweg–deVries equation and look at the *Variations* and *See also* sections.

- Implement some numerical scheme and demonstrate some properties of the KdV equation that you find interesting using it. There is a multitude of numerical methods that can be used for this and you can choose freely. A classical scheme is Zabusky–Kruskal [6] and a more recent review of a few different schemes can be found in [5]. Note that even if you use a software package written by someone else you need to describe the numerical scheme in your report. This can also be combined with a discussion of equations related to the KdV equation if you use a numerical scheme that can be used on the other equations as well.

- Construct the general exact solutions for the KdV equation requires some fairly sophisticated mathematics so this is also a suitable topic. You could for instance choose some interesting parts from [9] or [8] and describe in your report.

BCH and Reed–Solomon error-correcting codes

When sending information from one place to another there is always a risk that something goes wrong and part of the signal is distorted or lost. This is usually handled by making sure that the information that is being sent in *encoded* in a particular way that allows for either *error-detection*, the recipient can tell that the signal has disturbed and can ask for it to be resent, or *error-correction*, the recipient can restore a damaged signal and get the correct message.
There are many different codes that can used for error-detection and error-correction. The Hamming-codes are usually the most basic ones and one useful generalization of them are the Bose-Chaudhuri-Hocquenghem (BCH) codes. In this project you will discuss these codes or a commonly used subclass of the called the Reed–Solomon codes.
In this project you should

- Describe the basic concepts of codes, error-detection, error-correction, encoding and decoding. Give a basic description of how this is done with the BCH (or Reed–Solomon) codes.

and also do at least one of the following

- Give proofs for what errors the BCH (or Reed–Solomon) codes can detect and correct.
- Implement one of the codes and demonstrate that is works, you can find some inspiration for how to visualize the result of using an error-correcting code in [10].

Descriptions of the BCH codes can be found in most books on basic coding theory, see for example [11, 12]. You can also find a short introduction to information and coding theory on the publicly available Blackboard page for the *Quantum Computation and Information course*.

Approximation of curves using polynomial splines

In many applications is desirable to have a simple but controllable representation of curves and other geometry. One way to achieve this is to approximate curves with polynomials. To make complicated curves you will generally need high-order polynomials and this can be problematic and difficult to control (for example due to Runge's phenomenon). One solutions this is the concept of splines that are piecewise polynomial functions that are very cleverly defined using only a small set of points. Splines have been used in many areas such as computer aided design and manufacture, cinematic computer rendering and statistical regression.
In this project you should

- Describe the basic concepts of spline and describe how they are defined and calculated. It can be enough to focus on a particular class of splines, such as cubic B-splines, but this requires the rest of the report to be more extensive.

and also do at least one of the following

- Using some suitable software, most mathematical software that lets you plot things is enough, a fairly lightweight freeware option is GeoGebra, give a thorough visual and mathematical demonstration of how splines work. Both show how to find splines that interpolates a given set of points ans well as making some interesting shapes, star shapes and heart shapes are good examples. Try to create the shapes using as few splines and control points as possible.

- Describe how tensor product of splines can be used to approximate surfaces and show some examples.

- Splines can also be used together with other methods for analysing and approximating curves. One example of this is spline wavelets. Describe what spline wavelets are and their basic properties, see for example [15, 16].

Some good general sources for this project are [13, 14].

Solution of matrix differential equations

Differential equations are common modelling tools in many different areas. A simple type of models are linear systems of ordinary differential equations, that is systems of ordinary differential equations that can be written on the form

$$\mathbf{x}'(t) = A\mathbf{x}(t)$$

where A is a square matrix.
The form and expressions of the solutions to such a system depends on the eigenvalues of the matrix. The eigenvalues of the system also gives a lot of information about the long-term behaviour of the solutions.
In this project you should

- Describe the general method for solving a matrix differential equation and illustrate with examples of low dimensions (at least one example using a 3-dimensional of 4-dimensional system). You can use the method described in [17].

and also do at least one of the following

- Describe the concepts of stability, attractor and repulsor for a 2×2 matrix differential equation. Illustrate the concepts with phase portraits, see [18]. Try to explain how we can extend these ideas to higher dimensons.
- Matrix differential equations are systems of ordinary differential equations but there are similar methods for solving some systems of partial differential equations, see [17]. Give an example of this and describe the similarities between the methods.

References

[1] *Solving Least Squares Problems*, Lawson, C. L., Hanson, R. J., Prentice Hall Inc., Englewood Cliffs, New Jersey (1974)

[2] *Numerical methods for least square problems*, Å. Björck, SIAM, Philadelphia (1996)

[3] *A method for the solution of certain non-linear problems in least squares*, Levenberg, K., Quarterly Journal of Applied Mathmetics II:2 (1944), page 164–168

[4] *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Press, W. H., Teukolsky, S. A., Vetterling, W. T. and Flannery, B. P., Cambridge University Press, 3rd Edition, New York (2007)

[5] *Applying Explicit Schemes to the Korteweg-de Vries Equation*, M. Shahrill, M. S. F. Chong, H. N. H. M. Nor, Modern Applied Science, Vol 9, No. 4, (2015), page 200-224

[6] *Interaction of "solitons" in a collisionless plasma and the recurrence of initial states*, N. J. Zabusky and M. D. Kruskal, Physical Review Letters, Vol. 15, No. 6 (1965), page 240-243

[7] *Fluid Mechanics*, P. K. Kundu, I. M. Cohen, Academic Press, 4th Edition (2008)

[8] *Determinants and Their Applications in Mathematical Physics*, R. Vein, P. Dale, Springer-Verlag New York (1999)

[9] *Solitons*, P. G. Drazin, Cambridge University Press (1983)

[10] *Free space laser communication experiments from Earth to the Lunar Reconnaissance Orbiter in lunar orbit*, X. L. Sun *et. al.*, Optics Express, Vol. 21, No. 2 (2013) page 1865-1871

[11] *Introduction th the Theory of Error-Correcting Codes*, V. Pless, John Wiley & Sons Inc., (1998)

[12] *The Theory of Information and Coding*, R. J. McEliece, Cambridge University Press (2002)

[13] *A Practical Guide to Splines*, C. de Boor, Springer-Verlag New York (2001)

[14] *Curves and Surfaces In Geometric Modeling: Theory And Algorithms*, J. Gallier, Morgan Kaufmann Publishers (2000)

[15] *Ten Good Reasons for Using Spline Wavelets*, M. Unser, in *Proc. SPIE Vol. 3169, Wavelets Applications in Signal and Image Processing V* (1997), page 422-431

[16] *A Cardinal Spline Approach to Wavelets*, C. K. Chui, J-Z. Wang, Proceedings of the American Mathematical Society, Vol. 113, No. 3 (1991), page 785-793

[17] *Differential Equations: An Operational Approach*, H. M. Moya-Cessa, F. Soto-Eguibar, Rinton Press Inc. (2011)

[18] *Differential Equations, Dynamical Systems, and an Introduction to Chaos*, M. W. Hirsch, R. L. Devaney, S. Smale, Academic Press (2013)