104-th European Study Group with Industry (ESGI'104)

September 23 – 27, 2014
Sofia, Bulgaria

PROBLEMS
&
FINAL REPORTS


Demetra
2014

Organizers:

Institute of Information and Communication Technologies,
Bulgarian Academy of Sciences

Faculty of Mathematics and Informatics,
Sofia University "St. Kl. Ohridski"

Institute of Mathematics and Informatics,
Bulgarian Academy of Sciences

in cooperation with

European Consortium for Mathematics in Industry

Scientific Advisory Committee:

Prof. Dr.Sc. Svetozar Margenov, Director of IICT, BAS
Doc. Dr. Evgenia Velikova, Dean of FMI, SU
Prof. Dr.Sc. Julian Revalski, Director of IMI, BAS

Local Organizing Committee:

Prof. Dr.Sc. Stefka Dimova, FMI, SU
Doc. Dr. Krasimir Georgiev, IICT, BAS
Doc. Dr. Nadya Zlateva, FMI, SU
Doc. Dr. Neli Dimitrova, IMI, BAS
Doc. Dr. Veselin Gushev, FMI, SU
Dr. Ivan Georgiev, IICT and IMI, BAS

Published by Demetra Publishing house, Sofia, Bulgaria
ISBN 978-954-9526-87-5

Contents

Preface

The 104th European Study Group with Industry (ESGI'104) was held in Sofia, Bulgaria, September 23–27, 2014. It was organized by the Institute of Information and Communication Technologies, Bulgarian Academy of Sceinces (IICT-BAS), the Faculty of Mathematics and Informatics, Sofia University "St. Kliment Ohridski" (FMI–SU) and the Institute of Mathematics and Informatics, BAS (IMI–BAS) in cooperation with the European Consortium for Mathematics in Industry (ECMI). The ESGI'104 was the second Study Group in Bulgaria, after the very successful ESGI'95, September 23–27, 2013.

ESGI'104 was financially supported by the Sofia University grant N025/2014. The event was hosted by the Institute of Information and Communication Technologies, BAS, and by the Institute of Mathematics and Informatics, BAS. The two institutions provided excellent conditions for work.

Study Groups with Industry are an internationally recognized method of technology transfer between academic mathematicians and industry. These week long workshops provide a forum for industrial scientists to work alongside academic mathematicians on problems of direct industrial relevance.

Seven problems were selected by the Organizing Committee and proposed for working on:

1. *Effective recognition of video patterns in a recorded video stream*, CROSS Agency Ltd.;
2. *Relaxation of surface tension after a large initial perturbation*, KRÜSS GmbH;
3. *Smoothing of well rates in subsurface hydrocarbon reservoir simulators*, pp-Research Ltd.;
4. *Finding an effective metric used for bijective S-box generation by genetic algorithms*, State Agency for National Security;
5. *Cyber threats optimization for e-government services*, STEMO Ltd.;
6. *Effect of the precipitation of acid soap and alkanoic acid crystallites on the bulk pH*, Unilever Trumbull, Connecticut, USA;
7. *Circular arc approximation of pointwise curves for use in the NC programing*, Vintex Ltd.

Five of the companies are Bulgarian. The problems of the companies KRÜSS and Unilever were presented by one of their Bulgarian collaborators, Prof. Danov from the Faculty of Chemistry and Pharmacy, Sofia University.

Thirty seven participants from Bulgaria (32) and from abroad (5), divided into seven groups, worked on the problems. The Bulgarian participants were from different Academic institutions: FMI–SU; IMI–BAS; IICT–BAS; Plovdiv University; Veliko Tarnovo University; Technical University of Sofia. We were happy to have prof. Poul Hjorth from the Technical University of Denmark, organizer of Study groups in Denmark and member of the Council of ECMI, as a participant at ESGI'104. Three colleagues from the University of Štip, Macedonia and one from The Imperial College, London, made a valuable contribution to the work on the problems.

Six of the participants were students: two PhD, three Master, one bachelor student. The Master students were from the Master program Computational mathematics and mathematical modelling, evaluated as ECMI Master program in industrial mathematics, branch Technomatematics.

On the last day of the workshop the progress in treatment the problems and recommended routes forward were presented. On their basis the final reports of the groups are prepared to provide a formal record for both the industrial and academic participants. The reports are assembled in this booklet to form the Study Group Final Report.

The description of the problems, the last day presentations and the final reports of each working group are posted on the website of the ESGI'104:

`http://parallel.bas.bg/esgi104`

As at ESGI'95, certificates for participation and for valuable contribution were given to the participants.

It is worth mentioning that the event attracted Bulgarian media attention – Prof. Svetozar Margenov and Prof. Stefka Dimova were invited to give a half an hour interview for the National radio "Hristo Botev". On the next day a representative of a company turned to the organizers of ESGI for assistance in solving a problem. The visibility of this event in the scientific community and in the media is promising to deepen and expand the relationships between academia and industry. So the mission of the SGI is accomplished. The next Bulgarian Study Group is already planned for September 21–25, 2015.

List of participants

Aleksandra Stojanova (University of Štip, FCS)
Ana Avdzhieva (Sofia University, FMI)
Angela Slavova (IMI, BAS)
Atanas Nikolov (IICT, BAS)
Daniela Georgieva (Technical University of Sofia, FAMCS)
Dimo Dimov (IICT, BAS)
Doychin Boyadzhiev (Plovdiv University, FMI)
Dragomir Aleksov (Sofia University, FMI)
Dusan Bikov (University of Štip, FCS)
Georgi Ivanov (SANS)
Gergana Velikova (Sofia University, FMI)
Ivan Bazhlekov (IMI, BAS)
Ivan Georgiev (IMI, BAS/IICT, BAS)
Ivan Hristov (Sofia University, FMI)
Krassimira Ivanova (IMI, BAS)
Liliya Stoykova (Sofia University, FMI)
Limonka Lazarova (University of Štip, FCS)
Maroussia Bojkova (Sofia University, FMI)
Michail Todorov (Technical University of Sofia, FAMI)
Milena Veneva (Sofia University, FMI)
Miroslav Ivanov (IMI, BAS)
Nikola Simeonov (ppResearch Ltd.)
Nikolay Nikolov (SANS)
Nikolay Shegunov (Sofia University, FMI)
Ognyan Kounchev (IMI, BAS)
Pablo Crotti (Imperial College London)
Pencho Marinov(IICT, BAS)

Plamen Mateev (Sofia University, FMI)
Poul Hjorth (Technical University of Denmark)
Roumyana Yordanova (IMI, BAS)
Savka Kostadinova (Sofia University, FMI)
Stefka Dimova (Sofia University, FMI)
Stela Zhelezova (IMI, BAS)
Tihomir Ivanov (IMI, BAS)
Tsonka Baicheva (IMI, BAS)
Vasil Kolev (IICT, BAS)
Yuri Borisov (IMI, BAS)
Zlatogor Minchev (IICT, BAS/IMI, BAS)

PROBLEMS

Problem 1. Effective recognition of the video pattern in a recorded video stream

CROSS Agency Ltd.

Nikolay Ivanov

Information Agency CROSS is a source of information about the Bulgarian central and regional electronic and print media (newspapers, magazines, radio and television, online publications and agencies) for customers, government institutions and non-governmental organizations in Bulgaria and Europe.

One of the monitoring, made by CROSS Agency, is radio and TV clipping. CROSS Agency supports 24 hour record of 18 national and regional radio and TV programs. The agency's team makes a complete transcript of the 3 national radio and 4 national TV programs. All emissions on half hour and hour are deciphered. For all other radio or TV broadcasts the Agency provide records or full-text deciphering after customer query.

The media analysis of companies shows how the customer's company is represented in the media – who most represents the company, which experts comment company or its products, and also the same analysis for the competitive institutions in order to compare theirs publicity parameters with those of the customer's company.

One of the main parts of the bulletin, made for the customers, is quantitative and content analysis. The bulletin contains summary of the total number of materials, media types, placement of the material, customer's reference in the material and assessment of the attitude of materials.

The considerable part of the TV announcements of customer's company and/or competition companies is the broadcast of theirs advertisements. The advertisements are in advertisements' blocks, spread within other program elements. At that moment, the search of a particular advertisement is made by operator looking in the recorded material. The records are half-hour video portions in MP4 format.

The goal is to automate this process.

For facilitating of the task we can assume that for some period the advertisements of the costumer company and its competitive companies are fixed.

For the experiments the CROSS Agency gives samples of the advertisements of 3 Bulgarian banks and 4 hours record of one day of bTV broadcast.

This way, the task stands – **the effective search of the concrete video pattern in a recorded video stream**.

The expected results are connected with:

- discussing variants of solving the task – used descriptors, distance measures, etc.;
- presenting the entire algorithm for automation of the process;
- choice of appropriate software programs and tools for solving the stages of this algorithm;
- experiments and the discussion of the results.

Problem 2. Relaxation of surface tension after a large initial perturbation

KRÜSS GmbH – Germany, www.kruss.de

Krassimir Danov, www.lcpe.uni-sofia.bg

1. Introduction

KRÜSS GmbH – Germany is the world's leading supplier of measuring instruments for surface and interfacial tension, contact angles, foam analysis, interfacial rheology, etc. KRÜSS GmbH – Germany not only provide high quality product solutions – their offers are a combination of technology and scientific consulting. The adequate mathematical modeling of precise measured data makes the apparatuses attractive for many applications in the chemical and pharmaceutical industries, medicine, biotechnology, ecology, food and beverages production, etc. The problem for one component solution, which is described below, is the starting point for the intensive future developments. The fast and stable algorithms can be extended for: a) multi component systems; b) mixed barrier-diffusion control; c) ionic surfactants in the presence of salt; d) protein and polymer solutions, etc.

2. Mathematical formulation of the problem

The diffusion process of a simple one component solution is described by the following linear partial differential equation:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2} \quad \text{for} \quad t > 0 \quad \text{and} \quad x > 0 \tag{1}$$

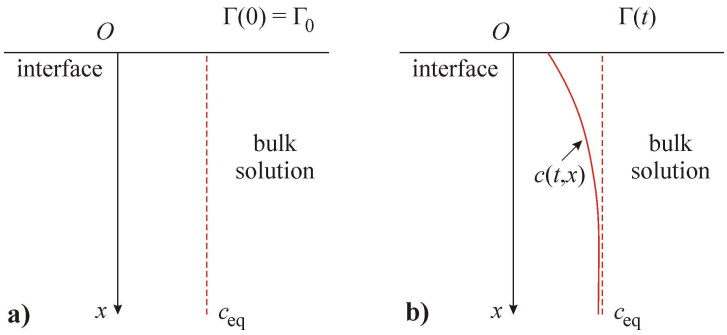


Fig. 1. Sketch of the diffusion problem: a) initial conditions; b) concentration profile at a given moment of time.

where x is the spatial coordinate, t is time, D is the diffusion coefficient, and $c(t,x)$ is the bulk concentration of surfactant (see Fig. 1). The input concentration in the solution is c_{eq} so that the initial condition and the boundary condition at large distances from the interface for Eq. (1) read:

$$c(0,x) = c_{eq} \text{ for } x > 0 \quad \text{and} \quad \lim_{x\to\infty} c(t,x) = c_{eq} \quad \text{for} \quad t \geq 0 \tag{2}$$

Table 1. Typical adsorption isotherms and surface equations of state

	Adsorption isotherm	Equation of state
Frumkin	$Kc_s = \dfrac{\theta}{1-\theta}\exp(-\beta\,\theta)$	$\dfrac{\sigma_0-\sigma}{E_B\Gamma_\infty} = -\ln(1-\theta) - \dfrac{\beta}{2}\theta^2$
Van der Waals	$Kc_s = \dfrac{\theta}{1-\theta}\exp\left(\dfrac{\theta}{1-\theta} - \beta\,\theta\right)$	$\dfrac{\sigma_0-\sigma}{E_B\Gamma_\infty} = \dfrac{\theta}{1-\theta} - \dfrac{\beta}{2}\theta^2$
Helfand, Frisch, Lebowitz	$Kc_s = \dfrac{\theta}{1-\theta}\exp\left[\dfrac{3\theta-2\theta^2}{(1-\theta)^2} - \beta\,\theta\right]$	$\dfrac{\sigma_0-\sigma}{E_B\Gamma_\infty} = \dfrac{\theta}{(1-\theta)^2} - \dfrac{\beta}{2}\theta^2$

The main difference from the classical diffusion problems arises because of the adsorption, $\Gamma(t)$, at the interface, $x = 0$. The change of the adsorption is compensated by the diffusion flux from the bulk:

$$\frac{\mathrm{d}\,\Gamma}{\mathrm{d}\,t} = D\frac{\partial\, c}{\partial\, x} \quad \text{for } t > 0 \quad \text{and } x = 0 \tag{3}$$

Eq. (3) plays a role of the boundary condition for the diffusion problem. At initial time, $t = 0$, the interface contains a given amount of surfactants, Γ_0, so that:

$$\Gamma(0) = \Gamma_0 \tag{4}$$

To close the problem (1)–(4) one needs a relationship between the subsurface concentration, $c_s(t)$, defined as

$$\lim_{x\to 0} c(t,x) = c_s(t) \text{ for } t \geq 0 \tag{5}$$

and the adsorption, $\Gamma(t)$. This relationship is called "the adsorption isotherm". Different surfactants obey different adsorption isotherms. Usually three typical adsorption isotherms are used (see Table 1), where K is the adsorption constant, β is the interaction parameter, Γ_∞ is the maximum possible adsorption, and θ is the surface coverage, given by the expression

$$\theta(t) \equiv \frac{\Gamma(t)}{\Gamma_\infty} \quad \text{and} \quad 0 \leq \theta(t) < 1 \tag{6}$$

Note that the adsorption isotherms are nonlinear equations so that the problem has not an analytical solution.

Mathematical problem: Solve the diffusion equation (1) with initial and boundary conditions (2)–(6) for a given adsorption isotherm (see Table 1) to obtain the change of adsorption with time, that is $\Gamma(t)$.

3. Application of the mathematical model for the characterization of surfactants

Available apparatuses measure indirectly the relaxation of adsorption. The most sensitive to the change of adsorption is the surface tension, $\sigma(t)$. Fig. 2 shows the measured relaxation of surface tension at air-solution interface for 0.1 mM SDS. One sees that all 6 different runs have excellent reproducibility.

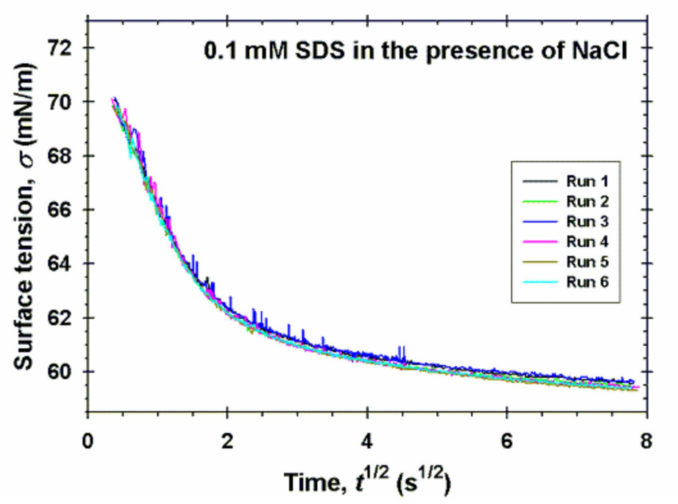


Fig. 2. Relaxation of surface tension $\sigma(t)$ at air-solution interface. The solution contains 0.1 mM SDS (sodium dodecyl sulfate) and 100 mM NaCl. The initial adsorption is zero and the temperature is 25 °C

The surface tension is related to the adsorption by the so called "2D equation of state". Different adsorption isotherms have different equations of state (see Table 1), where σ_0 is the surface tension without added surfactants and E_B is the Boltzmann energy.

Practical problem: Characterize the surfactant SDS from measured relaxation of surface tension (Fig. 2). That is, fit the experimental data using the van der Waals model (Table 1) to obtain the parameters K and Γ_∞.

Note that the addition of 100 mM NaCl is used to suppress the electrostatic interactions. NaCl is not a surface active substance and the concentration of NaCl is not included in the calculations. The values of the known parameters are:

$$\begin{aligned} &c_{eq} = 0.1 \ \text{ mol/m}^3,\ \Gamma_0 = 0 \text{ mol/m}^2,\ D = 5.5 \times 10^{-10} \ \text{ m}^2/\text{s},\ \beta = 1.62 \\ &\sigma_0 = 72.2 \ \text{ mN/m } = 0.0722 \text{ N/m},\ E_B = 2479 \ \text{ N} \cdot \text{m/mol} \end{aligned} \tag{7}$$

For convenience the surface tension measurements are presented in mN/m – for calculations one needs to use N/m, so that for example 70 mN/m (experimental data) corresponds to 0.07 N/m (for numerical calculations).

4. Ward-Torday integral equation

The numerical solution of the partial differential equation is time consumable. For that reason in the literature the following equivalent approach is used. The integral Laplace transform with respect to time of the function $f(t,x)$ is defined as follows:

$$F(s,x) = L[f] = \int_0^\infty f(t,x)\exp(-st)dt \tag{8}$$

where s is the parameter of the Laplace transform and $F(s,x)$ is the Laplace image. We introduce the Laplace images:

$$C(s,x) \equiv L[c(t,x) - c_{eq}]\ ,\ C_s(s) \equiv L[c_s(t) - c_{eq}] \tag{9}$$

The solution of the diffusion problem (1) with the initial and boundary conditions (2) reads:

$$C(s,x) = C_s(s)\exp\left(-x\sqrt{\frac{s}{D}}\right) \tag{10}$$

The Laplace transform of the boundary condition (3) with initial condition (4) gives:

$$sL[\Gamma - \Gamma_0] = D\frac{\mathrm{d}C}{\mathrm{d}x} \text{ at } x = 0 \tag{11}$$

From Eqs. (10) and (11) one derives

$$L[\Gamma - \Gamma_0] = -C_s\sqrt{\frac{D}{s}} \ \Rightarrow\ L[\Gamma - \Gamma_0] = -\sqrt{\frac{D}{s}}L[c_s(t) - c_{eq}] \tag{12}$$

Finally, applying the convolution theorem for Eq. (12) the Ward and Torday integral equation is obtained:

$$\Gamma(t) = \Gamma_0 - (\frac{D}{\pi})^{1/2} \int_0^t \frac{c_s(\tau) - c_{eq}}{(t-\tau)^{1/2}} \mathrm{d}\tau \tag{13}$$

Eq. (13) is the Volterra type equation, in which the kernel has a weak Abel type singularity.

Mathematical problem: Solve the integral equation (13) for a given adsorption isotherm (see Table 1) to obtain the change of adsorption with time, that is $\Gamma(t)$.

Problem 3. Smoothing of well rates in subsurface hydrocarbon reservoir simulators

ppResearch Ltd.

Peter Popov

Background

We are a start-up company which targets the hydrocarbon reservoir engineering community. We develop software tools for simulation of subsurface flows in deformable porous media. This is being done by modeling the physical problem by a system of partial differential equations. This system is discretized numerically and solved at a number of time steps. As a result of a simulation one has, as primary variable, the mechanical displacements and the fluid pressure resolved at any spatial location and time instance. A number of derived quantities, such as mechanical stresses, fluid velocity, etc are also computed. We are interested to relate such computations to localized processes which occur in particular locations in a subsurface reservoir.

The problem

A common problem in reservoir simulators is the history matching problem, where a number of wells are operated at a prescribed flow rate, measured by the operator. The data provides input to a simulator which then has to match various other measured quantities, such as pressure drop at wells, movement of saturation fronts, water break-out and other.

History matching computations are done multiple times until rock parameters are modified to achieve a satisfactory match. This requires a fast simulator. A common problem is that the input data is very rough and if input directly would cause considerable numerical difficulties, such as excessive Newton iterations to converge or excessively small time-steps.

A typical input for a well is a flow rate, specified at discrete time instances, which is positive at every instance. The goal is to replace the "rough" flow rate with a smoother function, which retains two properties of the original:

1. It remains positive at very instance;

2. The integral over the entire time range, i.e. the total liquid produced is preserved.

ppResearch will provide participants with typical datasets and would like to see different smoothing scenarios, using some characteristic time-scale which fulfill the above two properties.

Problem 4. Finding an effective metric used for bijective S-box generation by genetic algorithms

State Agency for National Security

Georgi Ivanov, Nikolay Nikolov

Background

In cryptography, the science of information protection, highly nonlinear mappings are wished for. Most of all encryption algorithms are built on the basis of nonlinear components, providing the cryptographic strength necessary to avoid any cryptanalytic attacks. Otherwise, the whole system breaking is equivalent to solving a linear system of equations. Usually, the only nonlinear components in symmetric encryption algorithms, specifically in block ciphers, are substitution tables or S-boxes: n-input m-output binary mappings. Among them, $(n \times n)$ bijective S-boxes are particularly interested. There are lots of methods know for S-box generation. The three main classes of such methods are: pseudo-random generation, algebraic constructions and various heuristic approaches. Among the latter are the genetic algorithms. Genetic algorithms are applied with the purpose to find the optimal solution of an optimization problem. Related to S-boxes, genetic algorithms aim at producing S-boxes that possess cryptographic properties which are optimal with respect to several targeted criteria simultaneously. Unfortunately, S-boxes that are optimal with respect to all desired criteria do not exist due to the criteria contradiction available. Thus, not optimal but sub-optimal S-boxes, which strength is still satisfactory, are needed and searched for.

The origin of the problem

The problem, stated below, has arisen in result of the application of a specific genetic algorithm in order to obtain strong bijective S-boxes (n-input n-output vectorial Boolean functions). The genetic algorithm is used in combination with a cost or a fitness function taken to ascertain which individuals will survive to the next generation. The cost function is based on the so called Walsh-Hadamard Transform Spectrum, which has to be flat in order S-boxes with good nonlinearity to be obtained. The fitness function only role is the S-box nonlinearity to be calculated. Bent S-boxes are of the highest nonlinearity possible (their Walsh spectrum is flat entirely – all spectral coefficients are equal to $2^{n/2}$) but they are

never balanced – something important and wished for. For that reason, S-boxes that are close to the Bent ones are needed (their cost, namely the difference between them and the Bent ones, is the smallest positive one possible).

The problem

The problem is related to finding any appropriate metric measuring the distance to an $(2^n \times 2^n)$ "flat" matrix of integer-valued elements, possessing equal absolute values of $2^{n/2}$, of two square matrices of equal dimensions $(2^n \times 2^n)$ with integer-valued elements and the sum of squares of all elements in each column is one and the same constant equal to 2^{2n}. The matrix, which is closer to the "flat" matrix with respect to the specified metric and different from it at the same time, is searched for.

Description:

Inputs: A, B and C matrices

Any two square matrices of equal dimensions with integer-valued elements, $A(m \times m)$ and $B(m \times m)$ respectively, where

(1) $m = 2^n$ for some integer $n > 7$; and

(2) the sum of squares of all elements in each column is one and the same constant equal to $m^2 = 2^{2n}$.

The optimal matrix C:

The optimal matrix with respect to our consideration is referred as the BENT matrix $C(m \times m)$ with "flat" integer-valued elements possessing equal absolute values $(= \sqrt{m} = 2^{n/2})$.

Searched output: matrix X

A sub-optimal matrix $X(m \times m)$ satisfying both of the conditions (1) and (2), which is as **closer** to the BENT matrix $C(m \times m)$ as possible (almost flat elements), and different from it at the same time (not all of the elements are the same).

The problem is related to the possible evaluation of both of the **distances**, between A and C, and B and C respectively, with the intention of picking the "flatter" matrix between A and B (the more **closer** to the BENT one).

Any metric (evaluation criterion), that is appropriate for measuring the distance between two such matrices and the BENT one, is needed. Hamming distance and Minkowski distance have already been tried.

Is there any other suitable?

It is necessary for the evaluation criterion:

1. To be sensitive enough, that is, to be able to detect any small changes in the matrix elements absolute values, which in turn adequately to bring an immediate effect on the ranking of the particular matrix evaluated.

2. To allow the maximal deviation, that is, the deviation between the matrix element of maximal absolute value and the value $2^{n/2}$ of all BENT matrix elements, to be detected as well.

Problem 5. Cyber threats optimization for e-government services

STEMO Ltd.

Veselin Politov

Problem description

The innovative role of e-government services is a part of nowadays technological progress in the digital society. STEMO Ltd. is also working in this field since 2008 with a number of successful stories.

Thanks to our experience, one of the key problems, concerning the topic, is the unforeseen emerging set of cyber threats. As these threats are related to both technologies and users, the uncertainties, concerning even short-time forecasts and resulting preventive measures, are quite demanding.

Being rather complex, the problem, at hand, requires experts' data combination with real observations of practically implemented e-government services.

As "optimization in general", concerning multiple risks, originating from future cyber threats is quite unreasonable, a multi-criteria risk matrix for "implemented technologies" and "digital society components of influence" could be defined.

Further on, the defined matrix is studied and optimized, regarding different risk parameters. The resulting multifaceted practical implementation is producing multiple risks' prognosis for future cyber threats severity.

Five key steps for solving the problem could be implemented:

1. Defining "implemented technologies" and "digital society components of influence" multiple cyber risks database.
2. Formulation of optimization models, concerning multiple cyber risks database.
3. Formulation of discrete optimization problems, taking into account the particular forecasting period.
4. Choosing a software environment for solving the formulated problems.
5. Numerical experiments and discussion of results.

Problem 6. Effect of the precipitation of acid soap and alkanoic acid crystallites on the bulk pH

Unilever – Trumbull, Connecticut, USA, www.unilever.com

Krassimir Danov, www.lcpe.uni-sofia.bg

1. Introduction

Unilever is one of the world's leading fast-moving consumer goods companies with products sold in over 190 countries. More than 2 billion consumers worldwide use a Unilever product on any given day. The Unilever turnover was 49.8 billion in 2013. More than 174000 people work for Unilever and more than 6000 people work in the global research and development centers (Trumbull, USA; Port Sunlight, UK; Colworth, UK; Vlaardingen, NL; Shanghai, China; Bangalore, India). The quality of the products is challenged with the significant scientific contribution from 7 Universities (Oxford, Cambridge, MIT, Nottingham, Liverpool, Jairpur, Sofia).

One of the strategies for controlling the bulk pH of the products is the usage of fatty acid salts. In the case of one component the mathematical problem is described below. In the products Unilever uses natural substances, which are multi component systems. The developed solution of the described problem can be extended for such kind of mixtures and the precise calculation of the solubility makes the mathematical model applicable for designing of complex materials with given physicochemical properties.

2. Mathematical formulation of the problem

As a basis we will use the theory of the pH of carboxylate soap solutions, which accounts for the presence of NaCl, NaOH, and CO_2. We denote: Z^- the alkanoate ion; M^+ the metal ion (Na^+); HZ is the non-dissociated alkanoic (fatty) acid; MZ is the non-dissociated neutral soap; OH^- is the hydroxyl ion; H+ is the hydrogen ion; HCO_3^- is the hydrogencarbonate ion, which appears because of the solubility of CO_2 from the atmosphere. Five basic equations express the dissociation equilibria of the fatty acid (HZ) and the neutral soap (MZ) molecules, the dissociation of water and CO_2, and the electro-neutrality of the solution:

$$c_{\mathrm{H}} c_{\mathrm{Z}} \gamma_{\pm}^2 = K_{\mathrm{A}} c_{\mathrm{HZ}} \quad \text{and} \quad c_{\mathrm{M}} c_{\mathrm{Z}} \gamma_{\pm}^2 = Q_{\mathrm{MZ}} c_{\mathrm{MZ}} \tag{1}$$

$$c_{\rm H} c_{\rm OH} \gamma_{\pm}^2 = K_{\rm W} \quad \text{and} \quad c_{\rm H} c_{\rm HCO_3} \gamma_{\pm}^2 = K_{\rm CO_2} \tag{2}$$

$$I = c_{\rm H} + c_{\rm M} = c_{\rm OH} + c_{\rm HCO_3} + c_{\rm Z} + c_{\rm A} \tag{3}$$

where c is the concentration of the respective compound, $c_{\rm A}$ is input concentration of salt (NaCl), I is the ionic strength, $K_{\rm W}$, $K_{\rm CO_2}$, $K_{\rm A}$ and $Q_{\rm MZ}$ are the respective dissociation constants. The activity coefficient, $\gamma_{\pm}$, for this kind of solutions is calculated from the semi-empirical formula:

$$\log_{10} \gamma_{\pm} = 0.055 I - \frac{0.5115\sqrt{I}}{1 + 1.316\sqrt{I}} \tag{4}$$

The amounts of the components M and Z incorporated in the solid phase (in the crystallites) per unit volume are given by the equations:

$$m_{\rm M} = c_{\rm T} + c_{\rm A} + c_{\rm B} - c_{\rm M} - c_{\rm MZ} \quad \text{and} \quad m_{\rm Z} = c_{\rm T} - c_{\rm Z} - c_{\rm HZ} - c_{\rm MZ} \tag{5}$$

where $c_{\rm T}$ is the input concentration of MZ and $c_{\rm B}$ is the input base concentration (NaOH).

The quality of the products depends considerably on the bulk pH, so that after the numerical solution of the problem one needs to calculate:

$$\mathrm{pH} = -\log_{10}(\gamma_{\pm} c_{\rm H}) \tag{6}$$

which is measured experimentally for a given composition.

2.1. Solutions with fatty acid precipitates. In this case the concentration of fatty acid is fixed and equal to the equilibrium solubility, $S_{\rm HZ}$. The amount of the component M incorporated in the solid phase is zero. Therefore, the system of equations is closed and

$$c_{\rm HZ} = S_{\rm HZ} \quad \text{and} \quad m_{\rm M} = 0 \tag{7}$$

2.2. Solutions with precipitate of j:n acid soap. If a precipitate of $(\mathrm{HZ})_j(\mathrm{MZ})_n$ acid soap is present, then one closes the system of equations with the following two conditions: a) mass balance of the amounts $m_{\rm M}$ and $m_{\rm Z}$ with the stoichiometry $(j:n)$:

$$\frac{m_{\rm M}}{n} = \frac{m_{\rm Z}}{n+j} \tag{8}$$

b) the solubility relation for a precipitate of $j:n$ acid soap:

$$c_{\rm H}^j c_{\rm M}^n c_{\rm Z}^{j+n} \gamma_{\pm}^{2j+2n} = K_{jn} \tag{9}$$

where K_{jn} is the respective solubility product.

General mathematical problem: Solve the polynomial equations in more than one variable

$$F_j(x_1, x_2, \ldots, x_N) = b_j \qquad (j = 1, 2, \ldots, N) \tag{10}$$

in which the coefficients depend slowly on the solution (x_1, x_2, ..., x_N) to obtain only the positive solution, that is

$$x_j > 0 \qquad (j = 1, 2, \ldots, N) \tag{11}$$

Note that the difference between concentrations (for example c_{H} and c_{M}) can be 10 orders of magnitude.

3. Application of the mathematical model for the characterization of precipitates

In the case of NaMy the values of the following constants are known:

$$K_{\mathrm{W}} = 6.81 \times 10^{-15}\ \mathrm{M}^2,\ K_{\mathrm{A}} = 1.995 \times 10^{-5}\ \mathrm{M},\ Q_{\mathrm{MZ}} = 2.84\ \mathrm{M} \tag{12}$$

Fig. 1 shows the experimental dependence of pH on the concentration, c_T, for NaMy solution without added salt and base, that is for

$$c_A = 0\ \mathrm{M} \text{ and } \ c_B = 0\ \mathrm{M} \tag{13}$$

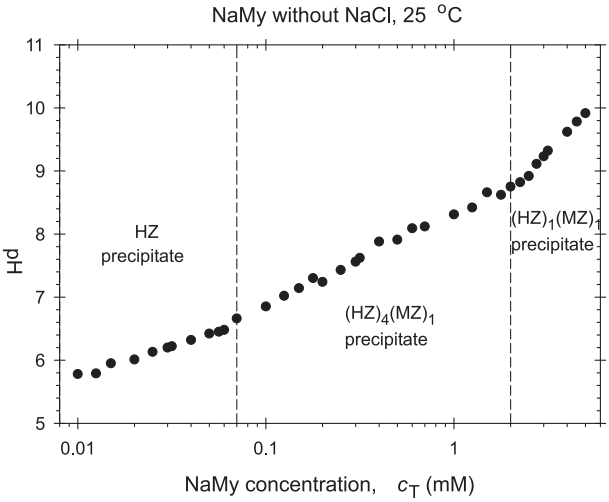


Fig. 1. Dependence of pH on the concentration of MZ (NaMy). Three different regions are measured: fatty acid precipitate; 4 : 1 precipitate; 1 : 1 precipitate

For convenience the experimental concentrations are given in mM, so that for example 10 mM (experimental value) corresponds to 0.01 M (for numerical calculations). The strategy of modeling is the following:

a) For concentrations below 0.07 mM one uses Section 2.1 (fatty acid precipitate) with the equilibrium solubility:

$$S_{\mathrm{HZ}} = 5.25 \times 10^{-7} \text{ M} \tag{14}$$

to fit the experimental data with one adjustable parameter, K_{CO_2}.

b) With the obtained value of K_{CO_2} one studies the concentration region from 0.07 to 2 mM. In this region the stoichiometry of the precipitate is known:

$$j = 4 \quad \text{and} \quad n = 1 \tag{15}$$

and one uses Section 2.2. The fit of experimental data gives the most probable value of the solubility product K_{41}.

c) Finally, for concentrations larger than 2 mM the stoichiometry of the precipitate is

$$j = 1 \quad \text{and} \quad n = 1 \tag{16}$$

and again the model described in Section 2.2 should be applied. From the fit of experimental data one obtains the solubility product K_{11}.

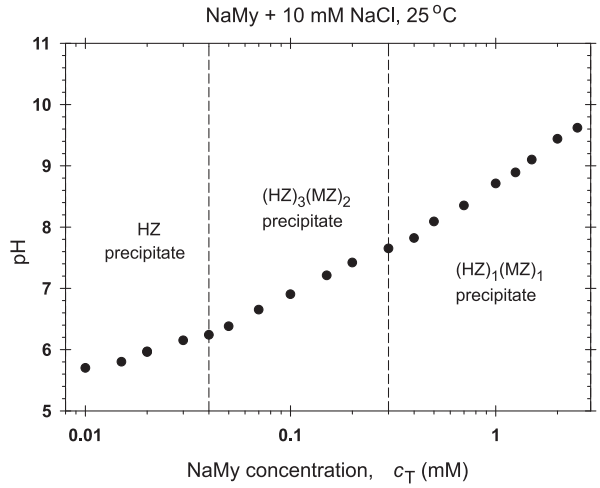


Fig. 2. Dependence of pH on the concentration of MZ (NaMy) in the presence of 10 mM NaCl. Three different regions are measured: fatty acid precipitate; 3 : 2 precipitate; 1 : 1 precipitate

Fig. 2 shows the experimental dependence of pH on the concentration, c_T, for NaMy solution in the presence of 10 mM NaCl:

$$c_{\mathrm{A}} = 0.01 \text{ M} \quad \text{and} \quad c_{\mathrm{B}} = 0 \quad \text{M} \tag{17}$$

In this case:

a) For concentrations below 0.04 mM HZ precipitate is observed. The fit of experimental data with the model from Section 2.1 gives K_{CO_2}.

b) For concentrations in the region from 0.04 to 0.3 mM the stoichiometry of the precipitates is $j : n = 3 : 2$, that is from the fit of experimental data with the model from Section 2.2 one obtains K_{32}.

c) Finally, for $c_{\mathrm{T}} > 0.3$ mM the stoichiometry of the precipitate is $j : n = 1 : 1$. Thus the most probable value of the solubility limit K_{11} can be obtained.

Note, that K_{CO_2} and K_{11} calculated from the two sets of experimental data (Figs 1 and 2) should be close each others.

Problem 7. Circular arc spline approximation of pointwise curves for use in the NC programing

Vintech Ltd.

Nikolay Spahiev

Context

Shapes used in the NC (numerical control) machine processing are created with lines and circular arcs. Often as a result of the high accuracy approximation of splines by CAD(computer aided design) systems, sets of thousands to millions of consequent points are generated, which result in huge NC programs, impossible for interpolation by the CNC (Computer numerical control) controllers of most machines.

Problem

A sequence of N ($N > 500$) points is given (X, Y:real). The points are connected in a polyline.

A new curve must be created from arcs and lines, such that:

– It passes through/nearby the given points in the same sequence.

– The distance from the new to the old curve does not exceed a given value E.

– It is composed of minimal number of elements.

Remarks

1. Lines and Circular arcs should be used.

2. Local minimum – fitting an arc to each set of 3 points – is not a solution of the task.

The participants will receive 2D sequences of points, in text format of the type:

$x1$, $y1$
$x2$, $y2$
$x3$, $y3$
...
xN, yN

The output should look like:

L $(xx1, yy1)$ $(xx2, yy2)$
A $(xx2, yy2)$ $(xx3, yy3)$ $(xxc, yyc) \pm 1$
Here (xxc, yyc) are the coordinates of the center of the circle;
$+1$: direction of the arc counterclockwise, -1: direction of the arc clockwise.
The coordinates of the output elements may or may not match those of the input points.

It is permissible to write the output in an SVG file for visualization.

The lengths of the sets will be of the magnitude of thousands of points, which excludes the possibility for bruteforcing.

Example1: $= 0.5$
0.4, 0
5.3, 0
9.6, 0
10, 0.3
10, 2.5
10, 3.9
9.7, 4.0
4.3, 4.0
0.2, 4.0
0, 3.7
0, 2.2
0, 0.1

Output:
L (0,0) (10,0)
L (10,0) (10,4)
L (10,4) (0,4)
L (0,4) (0,0)

Example 2: 500 points from an ellipse.

References

[1] R. L. Scot Drysdale, Gunter Rote, and Astrid Sturm. Approximation of an Open Polygonal Curve with a Minimum Number of Circular Arcs and Biarcs. Computational Geometry, Volume 41, Issues 1–2, October 2008, Pages 31–47. Special Issue on the 22nd European Workshop on Computational Geometry (EuroCG)

[2] Kazimierz Jakubczyk. Approximation of Smooth Planar Curves by Circular Arc Splines. May 30, 2010 (rev. January 28, 2012) http://www.kaj.pr.radom.pl/prace/Biarcs.pdf

[3] O. Aichholzer, F. Aurenhammer, T. Hackl, B. J ?uttler, M. Oberneder, and Z. Sır. Computational and structural advantages of circular boundary representation. http://www.industrial-geometry.at/uploads/nrn_report_98.pdf

FINAL REPORTS

Effective recognition of the video pattern in a recorded video stream

Atanas Nikolov, Dimo Dimov, Vassil Kolev,
Miroslav Ivanov, Krassimira Ivanova, Ognian Kounchev,
Maroussia Bojkova, Plamen Mateev

1. Introduction

The advertisements' detection and processing, as part of the media content analysis (MCA), has a number of uses and offers significant benefits to companies, organizations, different agencies and – particularly those that receive wide media coverage. MCA [1] is increasingly used commercially because of the key roles of the mass media. Fig. 1 provides an overview of the four roles and uses of MCA mainly within the two areas – strategic planning and evaluation. Since the advertisement TV block is media part of video stream and also a key in MCA, advertisements' detection and recognition is very important.

The media analysis of companies shows how the customer's company is represented in the media – which most represents the company, which experts comment company or the product, and also the same analysis for the competitive institution in order to compare their publicity parameters with those of the customer's com-

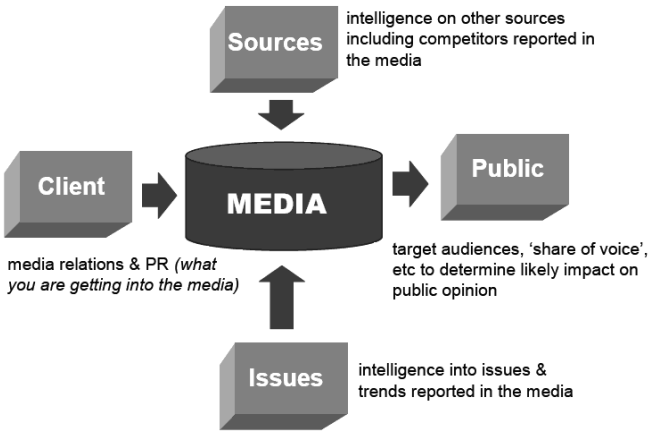


Figure 1: The four roles of media content analysis [1]

pany. The considerable part of the TV announcements of the customer's company and/or competition companies is the broadcast of theirs advertisements. The advertisements are broadcasted on television channels every day in advertisements' blocks, spread within other program elements (Fig.2). Companies pay a lot of money to place their advertisements on certain channels and in certain time slots. This way companies can assure that their advertisements were broadcasted.

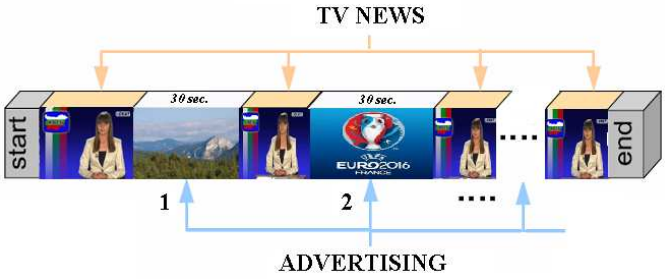


Figure 2: Example of a video stream with advertisements blocks

As it is mentioned in [2] there are some specifics of advertisements. Some of them are directly measurable, such as:

- repeated video sequence;
- restricted temporal length (generally between 10–60 sec);
- higher hard cut rate (scene changes);
- absence of correlation between consecutive scenes (due to camera and viewpoint change).

Other features are indirectly measurable, such as:

- high action rates (high motion & normalized difference energy);
- short shot lengths;
- drastically change of the visual style (like dominant colours and light);
- removal of the network-logo during advertisement blocks;
- turning up the volume of the audio signal during advertisements (in spite of the fact that some laws try to forbid this manner).

Another feature, mentioned in [2], connected to separating the consecutive advertisements by 5–10 blank frames, is no longer observed maybe because of better use of time for advertisements.

It should be noted that, the broadcasting of the advertisement can be transformed (usually by removing parts of it) depending on the viewer's preferences or

the TV time relevant. In [3] it is shown an example of detection and recognition of advertising trademarks from TV video stream of sport media.

2. Problem formulation

The problem can be formulated as follows:
Input:

1. A set of duration 30min video in MPEG4 format (25 fps; resolution 448x336 pixels), where the broadcast of 24 hours (or only prime time) of TV program is saved;
2. A set of advertisement video templates (about 30sec) in the same format.

Constrains:

The algorithms need to be appropriate with respect to time processing and hardware.

Output:

The time locations of a given advertisement template in the recorded TV stream, if this advertisement was broadcasted in the recorded day.

3. The team propositions

3.1. First scenario – Direct frame-by-frame pixel comparison

The simplest, but the slowest method for checking if a frame (image) of a given template video is contained in another video stream can be done by direct frame-by-frame pixel comparison of the two video sets. Thus, the maximum number of comparison will be $N_0 \left(\sum_{j=1}^{J} N_j \right)$, where N_0 is the number of frames in the video

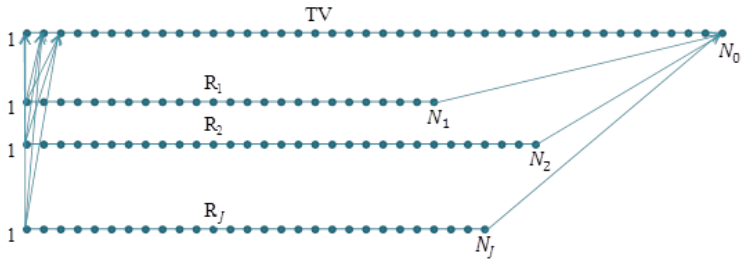


Figure 3: The direct frame-by-frame pixels comparison

stream; N_j is the number of frames in the j-th advertisement; J is the number of the observed advertisements. Fig. 3 shows the complexity of the algorithm. According to time processing this approach is not appropriate for us.

3.2. Second scenario – Speed-up, knowing JPEG's DCT coefficients in advance

Using DCT (Discrete Cosine Transform) coefficients of 8x8 region(s) from the JPEG frames, some acceleration in the frame comparison function could be achieved. This is true, because many of the DCT coefficients in 8x8 regions are zeros, and we will compare less numbers than pixels in the same region. In spite of this, the maximum number of comparison remains as in previous point. Furthermore, using DCT coefficients would have sense if we could extract them directly, but not computing them again.

3.3. Third scenario – Localizing the advertisement block and applying other scenarios in that block

The localization of the advertisements' blocks is very important in order that it solves more complex task – to focus the attention only on the frames of the advertisements blocks. This way, applying some more complicated algorithms only on this restricted area (searching for some objects as company's logo, slogan, some specific objects that are typical for the company or observed branch, etc.) one can find not only known, but also the new advertisements of the observed company.

In this case there can be analysed the place where the network logo is shown, taking into account the fact that during the advertisements blocks this logo is removed.

3.4. Fourth scenario – Matching the scenes duration of the advertisement and the TV stream

We stop our attention on the scenario, which decreases the complexity of the algorithm by replacing the frame-to-frame comparison with a scene-to-scene matching. This algorithm gives the advantages in case when the cardinality of the set of observed advertisements is bigger.

We represent the videos as 1D chain of numbers, which characterize the duration (in number of frames) of separate scenes of the videos. We consider a scene as a portion in a video where there is no sharp change between two consecutive frames. Thus, we compare the scenes duration belonging to the frames, and only for the suspicious scenes (which are able to match) we perform one-to-many frames comparison. This strategy has the big benefit of a much smaller number in-frame comparisons than previous cases.

4. Description of the proposed algorithm for the fourth scenario

Here we propose one simple and fast approach for scene detection analysing the difference between each two consecutive frames. This approach ensures detection of whole advertisements and/or arbitrary their pieces in a TV video stream.

Our algorithm consists of four steps:

1. Script the video streams as number vectors, representing the difference between neighbour frames (differential videos).
2. Split the stream to sequences of scenes and represent as number vector containing the durations (in frames) of each scene.
3. Comparison of number series (instead of frame series) between TV stream and potential candidates of scenes from the advertisement set.
4. In case of matching or inclusion of time interval from TV stream with time interval of some advertisement: frame comparison of representative frame from TV video with the frames in scene-candidate from the advertisement.

4.1. Forming differential videos (for TV stream and for advertisements)

The differential video can be estimated as:

$$V_{diff}(n) = \frac{1}{X \times Y} \sum_{\forall(x,y)} f\left(C_n(x,y), C_{n+1}(x,y)\right),$$

where

$n = 0, 1, \ldots, N-1$ are frame numbers in an observed video stream (TV record or advertisement);

$C_n(x,y)$ is the examined characteristic of a pixel (x,y) from the n-th frame. The examined characteristics can be luminosity, RGB colour, etc.;

$X \times Y$ is the frame size in pixels;

f is a function of the difference between $C_n(x,y)$ and $C_{n+1}(x,y)$. The function f can be defined in different ways.

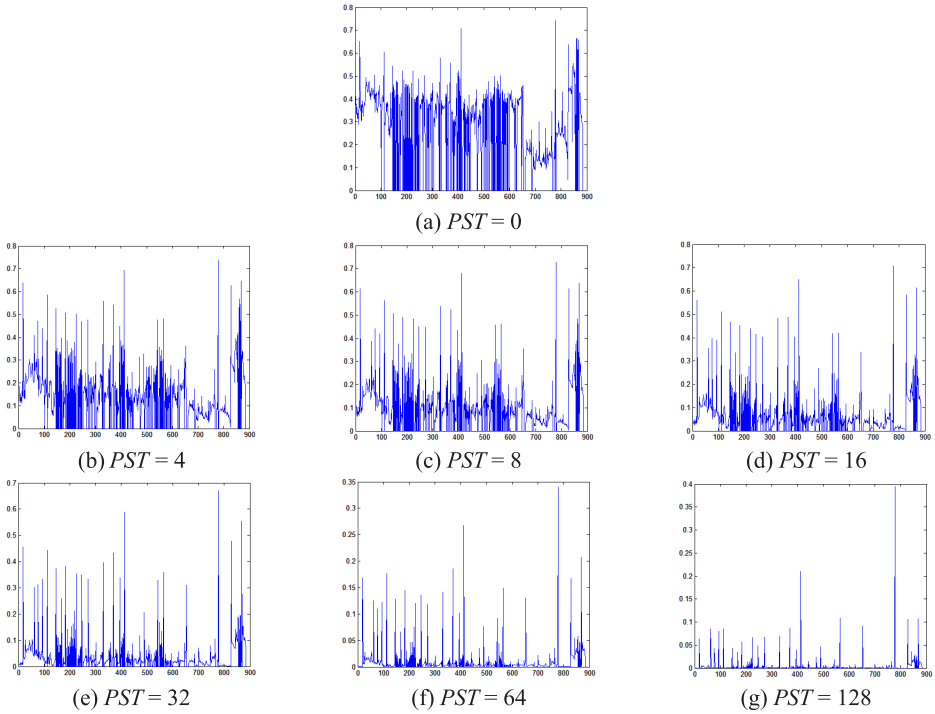
The simplest one is to calculate the absolute difference of the colour characteristics values:

$$f = \left|C_{n+1}(x,y) - C_n(x,y)\right|.$$

In order to get better true scenes separability from the in-scenes noise, a Pixel Similarity Threshold (PST) can be applied [6] [7]. This way the function f can be defined as:

$$f = \begin{cases} 1 & \text{if } \left|C_{n+1}(x,y) - C_n(x,y)\right| > PST, \\ 0 & \text{otherwise.} \end{cases}$$

Fig. 4 shows the alteration of the resulting sequence for different values of *PST*.



(a) $PST = 0$ (b) $PST = 4$ (c) $PST = 8$ (d) $PST = 16$ (e) $PST = 32$ (f) $PST = 64$ (g) $PST = 128$

Figure 4: The results from applying different values of the threshold PST

As we can see, when the threshold *PST* is too low, the considerable noise from the background is leaked into the foreground. On the other hand, when the threshold is too high, information from the foreground can be lost, since the system understands it as background. The objective is then to find such a threshold *PST* where the most information from the foreground pixels remains while the level noise is reduced.

Applying this algorithm on the TV video stream and on the advertisements, we receive:

- for the TV video stream: $(\mathrm{I}_{diff}(1), \ldots, \mathrm{I}_{diff}(n-1))$;
- for the j-th advertisement $(j = 1, \ldots, \mathrm{J})$, respectively: $(\mathrm{R}^{j}_{diff}(1), \ldots, \mathrm{R}^{j}_{diff}(n-1))$.

This algorithm for scene split guarantees the split in equal manner for the TV stream and for the advertisements.

4.2. Scenes detection and representation of videos as integer chains

The scene separation is connected with the process of finding the outliers in the integer chain obtained in the previous step (Fig. 5). The simplest approach to mark outliers is to choose a threshold thr_1, which is the same for the TV video and for the advertisement.

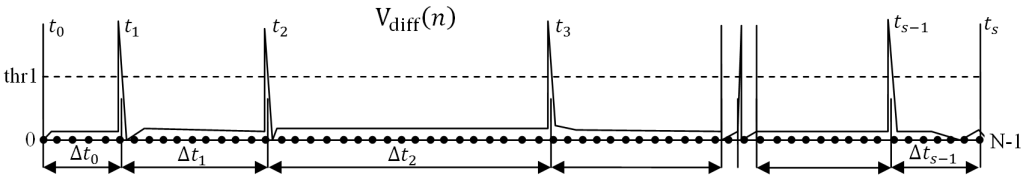


Figure 5: Visualisation of the process of scene detection

If $V_{diff}(n) > thr_1$ then the new scene starts $t_i \leftrightarrow n$; $\Delta t_i = t_{i+1} - t_i - 1$ is the length (in frame numbers) of i-th detected scene; $i = (0, \ldots, s-1)$; s is the number of scenes.

This way, the i-th scene is the chain of frames in the interval $[t_i; t_{i+1}] = [t_i; t_i + \Delta t_i]$.

The value of thr_1 can be obtained using different approaches. One possible variant is the statistical one based on the interquartile range ($thr_1 = Q_3 + 1.5(Q_3 - Q_1)$ [6]) of the learning sample. There are different datasets that can be used for this purpose:

1. to give as a learning set the advertisement stream. This would be good in case if we search only for one advertisement, because the calculation of thr_1 will be fast (this stream is no more than a minute) and also will take into account the exact specifics of the observed advertisement. For searching a set of advertisements this will lead to recalculating of the TV-chain for each advertisement, which will extremely slow up the process;
2. to give as a learning set the concrete observed video stream. This approach is also not good, because it is a slow process from one side, and all the advertisements also have to be recalculated each time from the other side;
3. to determine once on a training sample containing streams from various times of day and different TV channels. Because of the specifics of the advertisements, already mentioned in [2], this approach also is not so good;

4. to determine once using as a training sample a set of advertisements that are broadcasted in a certain time by different TV channels. This way the threshold will be obtained at once and will take into account more precisely the specifics of the advertisements.

After applying this algorithm on the TV video and on the advertisements we receive a set of chains:

- the TV chain: $I_{chain} = (\Delta t_0, \Delta t_1, \ldots, \Delta t_{s-1})$,
- the advertisements chains: $R^j_{chain} = (\Delta \tau^j_0, \Delta \tau^j_1, \ldots, \Delta \tau^j_{s_j-1})$,

where Δt_i, $i = 1, \ldots, s-1$ are the durations (in number of frames) of respective consecutive scenes in the TV stream and $\Delta \tau^j_k$, $k = 1, \ldots, s_j - 1$ is the similar but for the j-th advertisement stream, $j = 1, \ldots, J$.

Below we show the results of one experiment of scene detection and the creation of time-interval chain for 20 sec advertisement and for 15 min TV stream.

The function that determines the absolute difference of pixels (x, y) between n-th and $n+1$-th frame is:

$$f = |R_{n+1}(x,y) - R_n(x,y)| + |G_{n+1}(x,y) - G_n(x,y)| + |B_{n+1}(x,y) - B_n(x,y)|,$$

where R, G, B are resp. Red, Green, Blue values of the corresponding pixels.

The threshold thr_1 is calculated as IQR-outlier boundary using as a learning set the advertisement.

Figures 6 and 7 show the results of scene detection and creation of time-interval chain for 20 sec advertisement.

Note, that the breaks of the first frames into separate scenes is due to the fact that in the beginning of the advertisement the frames include moving of objects and background simultaneously, which leads to increasing the difference between neighbour frames.

Figures 8 and 9 show the result of scene detection and time-interval chain for 15 min TV stream of TV talk show. It is widely seen the increasing of the intensity of scene changes in the advertisement block.

The work of obtaining optimal value of thr_1 (constant or variable) has to continue in order to overcome splitting into too low scenes. But the algorithm must keep the property to split the scenes in equal manner in TV video and in the advertisements. Only this way we can reduce the task to the algebraic one.

4.3. Detection of potential matching or inclusion of scenes from TV in advertisements

In order to escape the damages of the length of the first and the last time intervals (because of the noising from neighbour broadcasts) they are excluded

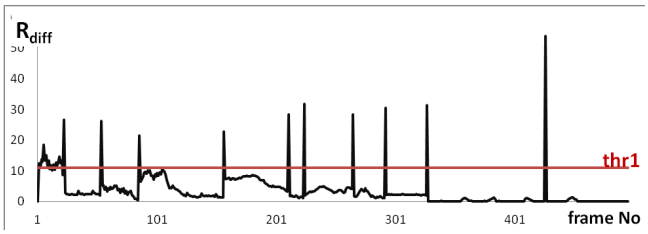


Figure 6: Scene detection of 20 sec advertisement

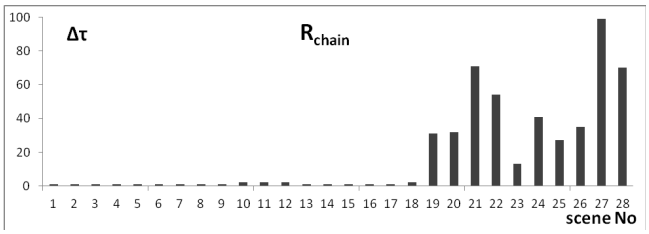


Figure 7: The time-interval chain that represents scenes duration for this advertisement

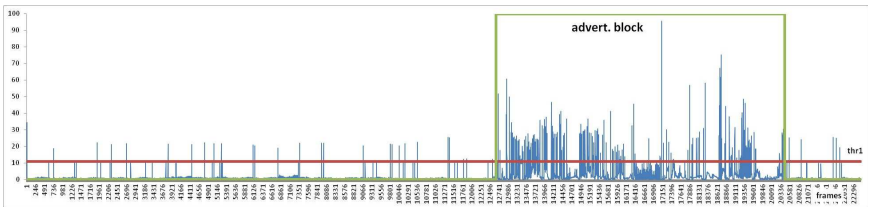


Figure 8: Scene detection of 15 min TV stream with one advertisement block

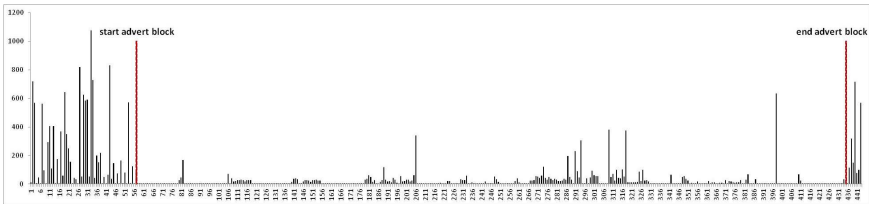


Figure 9: The time-interval chain that represents scenes duration for this TV stream

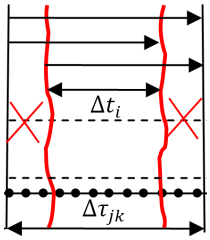


Figure 10: Cases of matching/inclusion of a scene from TV stream to a scene from the advertisement

from the observation. As it was mentioned in [2], one of the specific features of the advertisements is higher hard cut rate, which assures the bigger probability R^j_{chain} to be unique [7].

Main rule: One scene from a TV video can be part of a scene in a given advertisement (i.e. to match or to be shorter), while the inversed situation is not always valid.

This determines the searching direction for scenes comparison from a video stream to the set of advertisements.

The necessity of this rule arises because of the cases when we are given only the full advertisement template, but not the shorter variants, which consist of part(s) of the whole scenes $\Delta\tau^j_i$ from the full advertisement.

Usage of shorter variants is a regular practice of including the advertisements in the TV stream due to duration limitations, prices, thematic limitations, number of already made broadcasts of this advertisement, etc.

Thus, we can define four possible cases of inclusion of the scene Δt_k from video broadcast:

1. The Δt_k scene exactly matches with the $\Delta\tau^j_i$ scene from j-th advertisement;
2. Only the beginning of the Δt_i scene matches with the beginning of $\Delta\tau^j_k$ scene;
3. Only the end of the Δt_i scene matches with the end of $\Delta\tau^j_k$ scene;
4. The beginning and the end of Δt_i scene are internal for the $\Delta\tau^j_k$ scene. The Algorithm is applicable also for the cases when more than one scene from video is internal for a given scene from the advertisement.

Shortly the algorithm can be explained as:

Scan the scenes from TV stream consecutively ($\Delta t_i, i = 1, \ldots, s-1$)
If a previous scene was recognized as a scene from the j-th advertisement
Scan from the current scene $\Delta\tau_k^j$ for possible inclusion (comparing scenes duration)
If yes: continue with frame-to-frame comparison for adopting/rejecting hypothesis for finding a scene from the j-th advertisement (algorithm is explained in the next step)
If no: start searching for other advertisement comparing from the beginning of the scenes of each advertisement from the set
If a previous scene was not recognized as a scene from the j-th advertisement
Comparing Δt_i with scenes $\Delta\tau_k^j$ traversing all $j = 1, \ldots, J$ and $k = 1, \ldots, s_j - 1$ until find some scene as possible candidate or exhaust the scenes from the advertisements set.
If there was scene-candidate: continue with frame-to-frame comparison for adopting/rejecting hypothesis for finding a scene from the j-th advertisement.

4.4. Comparison of a frame from observed TV video scene with the frames in scene-candidate from the advertisement (if it exists)

This part is applicable when in the previous step the algorithm extracts the scene from the advertisement to be potentially matching or covering the observed scene from the TV stream.

Let:

- i be the index of observed scene from TV stream and k_j is k-th scene of the j-th advertisement that was nominated as candidate;
- f_1 and $f_{\Delta t_i}$ be respectively first and last frame from the i-th scene of TV stream;
- φ_x, $x = 1, \ldots, \Delta\tau_k^j$ be a frame from the scene-candidate (k) from the j-th advertisement.

In order to cover easily all possible variants of scene inclusion we make the frame comparison in the following manner (Fig. 11):

1. start with comparison of $f_1 = \varphi_1$
2. if yes – it covers first variant or second variant of inclusion
3. if no – start with comparison of $f_{\Delta t_i} = \varphi_{\Delta\tau_k^j}$
4. if yes – it covers the third variant of inclusion (the first one have to be already detected)
5. if no – consecutively continue comparison of: a) f_1 with the next frame from the beginning of the advertisement scene (inc(x):1,...,y) and b) $f_{\Delta t_i}$ with the previous scene from the end of the advertisement scene (dec(y):$\Delta\tau_k^j$,...x) until one of two situations arises:
   a. the equal frames are found – i.e. the scene of TV video belongs to the advertisement;
   b. two counters x and y meet each other – i.e. there was no frame in the advertisement scene equal to the first or last frame of the TV video.

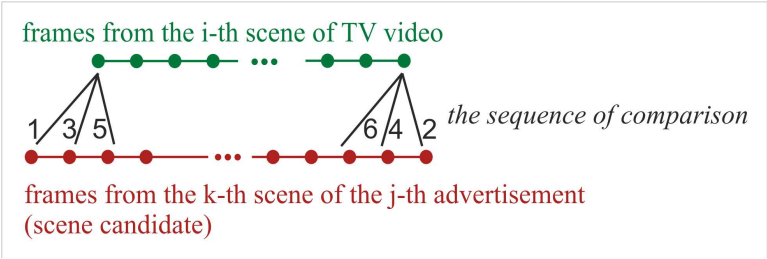


Figure 11: The sequence of frames comparison between scene from TV stream and scene-candidate from the advertisement

Conclusions

The observed approach ensures detection of the advertisements in a TV video stream for appropriate time processing and storage requirements.

MATLAB proved to be an important tool when developing prototypes due to its built-in video processing and mathematical tools. For real time implementation the use of lower level languages is required.

The first experiments of the program, realized on C#, showed the promised results.

References

[1] Macnamara J. Media content analysis: its uses, benefits and best practice methodology. In: Asia Pacific Public Relations Journal, vol. 6, no. 1, pp. 1–34, 2005.

[2] Tanwer, A. and P. S. Reel. Effects of threshold of hard cut based technique for advertisement detection in TV video streams. In: Proc. of the 2010 IEEE Students Technology Symposium, 03–04 April 2010, Kharagpur, India. DOI: 10.1109/TECHSYM.2010.5469157

[3] Lamberto B., B. Marco, and J. Arjun. A system for automatic detection and recognition of advertising trademarks in sports videos. In: Proc. of the 16th ACM Int. Conf. on Multimedia, pp. 991–992, 2008.

[4] Marcenaro, L., G. Vernazza, C. S. Regazzoni. Image stabilization algorithms for video-surveillance applications, IEEE Int. Conf. on Image Processing, 2001, Vol. 1, pp. 349–352.

[5] Dimov, D., A. Nikolov. Real time video stabilization for handheld devices, In: Rachev, B., A. Smrikarov (Eds.) Proceedings of CompSysTech'14, June 27, 2014, Ruse, Bulgaria, also in 2014 – ACM International Conference Proceeding Series, (to appear)

[6] Han, J. and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufman Publisher, Elsevier, 2006.

[7] Cover, T., J. Thomas. Elements of Information Theory, 2nd ed., John Wiley and Sons, 2006.

Relaxation of surface tension after a large initial perturbation

Ivan Bazhlekov, Stefka Dimova, Poul Hjorth,
Tihomir Ivanov, Angela Slavova, Roumyana Yordanova

Introduction

For convenience, we restate here the mathematical problem we have to solve (see the problem description in the first part of the booklet).

We have the diffusion equation

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2},\ t > 0,\ x > 0, \tag{1}$$

that describes the diffusion process in a simple one component solution, with initial condition

$$c(x,0) = c_{eq},\ x > 0, \tag{2}$$

right boundary condition

$$\lim_{x\to\infty} c(x,t) = c_{eq},\ t \geq 0, \tag{3}$$

and boundary condition at $x = 0$

$$\lim_{x\to 0} c(x,t) = c_s(t),\ t \geq 0, \tag{4}$$

where $c(x,t)$ is the bulk concentration of surfactant, $c_s(t)$ is the subsurface concentration. The latter is defined by a relation with the adsorption $\Gamma(t)$ at the interface, $x = 0$. This relation is called "the adsorption isotherm". Different surfactants obey different adsorption isotherms. Three of the most common ones are given in Table 1, where K is the so-called adsorption constant, β is the interaction parameter, Γ_∞ is the maximum adsorption, and θ is the surface coverage, given by $\theta(t) \equiv \Gamma(t)/\Gamma_\infty$.

Further, for the adsorption the following holds true:

$$\frac{d\Gamma}{dt} = D\left.\frac{\partial c}{\partial x}\right|_{x=0},\quad t > 0, \tag{5}$$

$$\Gamma(0) = \Gamma_0. \tag{6}$$

Table 1: Typical adsorption isotherms

	Adsorption isotherm
Frumkin	$Kc_s = \dfrac{\theta}{1-\theta}\exp\left(-\beta\theta\right)$
Van der Waals	$Kc_s = \dfrac{\theta}{1-\theta}\exp\left(\dfrac{\theta}{1-\theta} - \beta\theta\right)$
Helfand, Frisch, Lebowitz	$Kc_s = \dfrac{\theta}{1-\theta}\exp\left(\dfrac{3\theta - 2\theta^2}{(1-\theta)^2} - \beta\theta\right)$

Table 2: Typical surface equations of state

	Equation of state
Frumkin	$\dfrac{\sigma_0 - \sigma}{E_B\Gamma_\infty} = -\ln(1-\theta) - \dfrac{\beta}{2}\theta^2$
Van der Waals	$\dfrac{\sigma_0 - \sigma}{E_B\Gamma_\infty} = \dfrac{\theta}{1-\theta} - \dfrac{\beta}{2}\theta^2$
Helfand, Frisch, Lebowitz	$\dfrac{\sigma_0 - \sigma}{E_B\Gamma_\infty} = \dfrac{\theta}{(1-\theta)^2} - \dfrac{\beta}{2}\theta^2$

The difficulty in solving the problem (1)–(6), however, is in the fact that two of the parameters, namely K and Γ_∞, cannot be measured and, thus, are not known. So **our task is to find an algorithm for estimating the values of those two parameters**. For doing so, we are given experimental data for the interfacial tension σ (see Fig. 1), which is related to the other variables and parameters by the so-called "equation of state" (see Table 2).

The algorithms used for parametric identification are iterative [2, 3]. We begin with an initial estimate for the unknown parameters and then proceed by obtaining successive estimations that should converge to the real values. Those algorithms **rely on the ability to solve the differential problem efficiently**, if we know an estimate for the parameters. Thus, we begin our study with solving the differential problem. For this purpose, we propose four different numerical methods. Then, we explain how we can estimate the two unknown parameters.

Numerical Methods for Solving the Differential Problem

We suggest several different methods for solving the differential problem (1)–

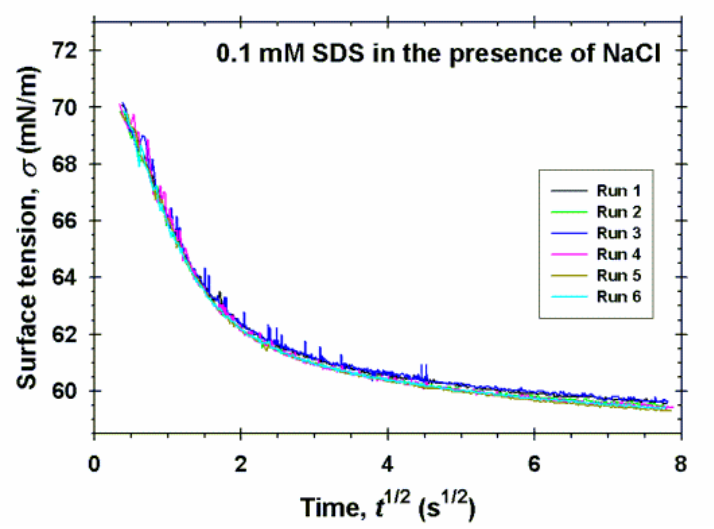


Figure 1: Experimental data for the interfacial tension

(6), that are explained below. Making numerical experiments, we compare them in terms of computational time.

Explicit Difference Scheme–1. Our first approach is to construct a more or less standard explicit difference scheme. In the set $\bar{\Omega} := [0, X] \times [0, T]$, we introduce a uniform mesh $\bar{\omega}_{h\tau} = \bar{\omega}_h \times \bar{\omega}_\tau$, where $\bar{\omega}_h := \{x_i = ih, i = \overline{0, n}, n = X/h\}$, $\bar{\omega}_\tau := \{t_j = j\tau, j = \overline{0, m}, m = T/\tau\}$.

We construct an explicit difference scheme in the following way. For the diffusion equation (1) we use the finite difference approximations

$$\frac{\partial c}{\partial t} \approx \frac{c(x, t+\tau) - c(x,t)}{\tau} \text{ and } \frac{\partial^2 c}{\partial x^2} \approx \frac{c(x+h,t) - 2c(x,t) + c(x-h,t)}{h^2}.$$

We obtain the difference equations

$$\frac{c_i^{j+1} - c_i^j}{\tau} = D.\frac{c_{i+1}^j - 2c_i^j + c_{i-1}^j}{h^2}, \ i = \overline{1, n-1}, j = \overline{0, m-1}.$$

The initial condition (2) and the right boundary condition (3) are approximated exactly:

$$c_0^0 = 0, \quad c_i^0 = c_{eq}, \quad i = \overline{0, n}.$$

For the left boundary condition (4) we have

$$c_0^{j+1} = c_s(t_j), \ j = \overline{0, m-1}.$$

Approximating (5), after the rescaling $\theta(t) \equiv \Gamma(t)/\Gamma_\infty$, we straightforwardly obtain

$$\theta^{j+1} = \theta^j + \frac{D\tau}{h\Gamma_\infty}(c_1^{j+1} - c_0^{j+1}),\ j = \overline{0, m-1}.$$

For the sake of completeness we also include an approximation for the interfacial tension $\sigma(t)$, using the second row of Table 2.

$$\sigma^{j+1} = \sigma_0 - E_B\Gamma_\infty \left[\frac{\theta^{j+1}}{1-\theta^{j+1}} - \frac{\beta}{2}(\theta^{j+1})^2\right],\ j = \overline{0, m-1}.$$

For all other equations of state we proceed analogously.

Explicit Difference Scheme–2. The second finite difference scheme uses nonuniform mesh in space. The spatial step h_i increases as a geometric progression with ratio q:

$\bar{\omega}_h := \{x_{i+1} = x_i + h_i, x_0 = 0, i = \overline{0, n}; h_{i+1} = h_i * q, i = \overline{2, n-3}\}$, with an exception that the first 3 and the last 2 steps are constant ($h_0 = h_1 = h_2$ and $h_{n-1} = h_n$). Thus keeping the ratio q close to 1 the mesh is locally almost uniform. The time step is constant as in the previous scheme.

In the tests performed here $h_0 = 2.5 * 10^{-7}$ and $h_n = 2.5 * 10^{-4}$ for $q = 1.2$, $n = 40$ and the time step is $\tau = 5 * 10^{-5}$.

In Scheme–2 we construct third order finite difference approximation of the spatial terms $\frac{\partial^2 c}{\partial x^2}$:

$$\frac{\partial^2 c}{\partial x^2} \approx a_1^i.c_{i-2}^j + a_2^i.c_{i-1}^j + a_3^i.c_i^j + a_4^i.c_{i+1}^j + a_5^i.c_{i+2}^j.$$

The coefficients $a_1^i = u_1, a_2^i = u_2, a_3^i = -(u_1+u_2+u_3+u_4), a_4^i = u_3, a_5^i = u_4 (i = \overline{1, n-1})$, where the vector $\mathbf{u}$ is the solution of the algebraic system:

$$\begin{array}{rrrrl}
-(h_{i-1}+h_i)u_1 & +(h_{i-1}+h_i)^2u_2 & -(h_{i-1}+h_i)^3u_3 & +(h_{i-1}+h_i)^4u_4 & = 0 \\
-h_iu_1 & +h_i^2u_2 & -h_i^3u_3 & +h_i^4u_4 & = 2 \\
h_{i+1}u_1 & +h_{i+1}^2u_2 & +h_{i+1}^3u_3 & +h_{i+1}^4u_4 & = 0 \\
(h_{i+1}+h_i)u_1 & +(h_{i+1}+h_i)^2u_2 & +(h_{i+1}+h_i)^3u_3 & +(h_{i+1}+h_i)^4u_4 & = 0
\end{array}$$

A third order approximation of (5) (applying the rescaling used in the previous scheme) is:

$$\theta^{j+1} = \theta^j + \frac{D\tau}{h\Gamma_\infty}(11/6c_0^{j+1} - 3c_1^{j+1} + 3/2c_2^{j+1} - 1/3c_3^{j+1}),\ j = \overline{0, m-1}.$$

The other elements of the scheme are as in the previous Scheme–1.

Ward and Torday Integral Equation. As it is shown in the problem description, the differential problem (1)–(6) is equivalent to the Integral Equation of Ward and Torday. Here we will show this by using a different approach.

Denote now by $c(x,t)$ the bulk concentration minus the equilibrium value c_{eq}. For the time evolution we then have the following problem:

$$\begin{aligned} \partial_t c(x,t) &= D\partial_x^2 c(x,t) \\ c(x,0) &= 0 \\ c(\infty,t) &= 0 \\ c(0,t) &= c_s(\theta(t)) - c_{eq} \end{aligned} \tag{7}$$

The last equation is in terms of a function f, which is the adsorption isotherm.

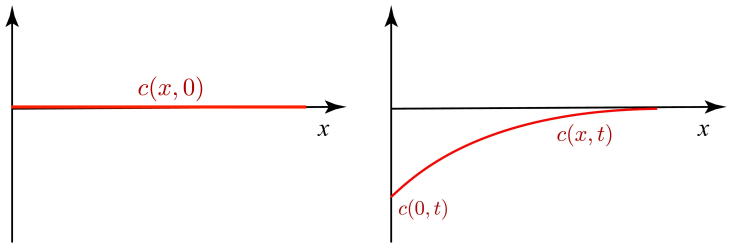


Figure 2: The evolution of the bulk concentration minus its initial value, c_{eq}. The surface adsorption, $\Gamma(t)$ grows in proportion to the diffusion flux from the bulk: $\mathrm{d}\Gamma(t)/\mathrm{d}t = D\ \partial_x c(x=0,t)$

Table 1 lists three separate adsorption isotherms, the Frumkin, the Van der Waals, and the Helfand-Frisch-Lebowitz isotherms.

The adsorption changes with time in proportion to the concentration gradient at the surface:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{D}{\Gamma_\infty}\partial_x c(x=0,t) \tag{8}$$

We will write down an expression for the solution to eq. (7), and then subsequently for the solution to eq. (8). We will initially treat $f(\theta(t))$ as a prescribed function, and then find a self consistent solution.

For the problem (7) we apply *Duhamel's Theorem*, which states that if $\Phi(x,t,\tau)$

denotes the solution to the auxiliary problem

$$\begin{aligned} \partial_t \Phi(x,t,\tau) &= D\partial_x^2 \Phi(x,t,\tau) \\ \Phi(x,0,\tau) &= 0 \\ \Phi(\infty,t,\tau) &= 0 \\ \Phi(0,t,\tau) &= c_s(\theta(\tau)) - c_{eq} \end{aligned} \tag{9}$$

where the right hand side in the last equation is taken to be a constant depending on a parameter τ rather that on t, then the solution $c(x,t)$ to the actual problem is given by

$$c(x,t) = \int_{\tau=0}^{\tau=t} \partial_t \Phi(x, t-\tau, \tau) \mathrm{d}\tau \tag{10}$$

The solution of the auxiliary problem (9) is given by

$$\Phi(x,t,\tau) = \frac{2(c_s(\theta(\tau)) - c_{eq})}{\sqrt{\pi}} \int_{\frac{x}{\sqrt{4Dt}}}^{\infty} \exp(-\eta^2) \mathrm{d}\eta$$

The partial derivative in (10) evaluates to

$$\partial_t \Phi(x,t,\tau) = (c_s(\theta(\tau)) - c_{eq}) \frac{x}{\sqrt{4\pi D}(t-\tau)^{3/2}} \exp\left[-\frac{x^2}{4D(t-\tau)}\right]$$

Substituting this into (10), we get

$$c(x,t) = \frac{x}{\sqrt{4\pi D}} \int_{\tau=0}^{\tau=t} \frac{(c_s(\theta(\tau)) - c_{eq})}{(t-\tau)^{3/2}} \exp\left[-\frac{x^2}{4D(t-\tau)}\right] \mathrm{d}\tau$$

We thus have

$$\partial_x c(x=0,t) = \frac{1}{\sqrt{4\pi D}} \int_{\tau=0}^{\tau=t} \frac{(c_s(\theta(\tau)) - c_{eq})}{(t-\tau)^{3/2}} \mathrm{d}\tau$$

The change in adsorption is given by:

$$\frac{\mathrm{d}\theta(t)}{\mathrm{d}t} = \frac{D}{\Gamma_\infty} \partial_x c(x=0,t) = \frac{1}{2\Gamma_\infty} \left(\frac{D}{\pi}\right)^{1/2} \int_{\tau=0}^{\tau=t} \frac{(c_s(\theta(\tau)) - c_{eq})}{(t-\tau)^{3/2}} \mathrm{d}\tau$$

Integrating with respect to t, we arrive at the Ward and Torday integral equation

$$\theta(t) = \theta_0 - \frac{1}{\Gamma_\infty} \left(\frac{D}{\pi}\right)^{1/2} \int_0^t \frac{(c_s(\theta(\tau)) - c_{eq})}{(t-\tau)^{1/2}} \mathrm{d}\tau$$

which is equation (13) in the problem description.

As a first option for solving numerically the Ward and Torday integral equation for $t = t_j = j\tau, j = 0, 1, \ldots$ we applied **the method of quadratures by using a modification of the Left Rectangle Rule**:

$$\theta(t_{j+1}) = \frac{\Gamma_0}{\Gamma_\infty} - \frac{1}{\Gamma_\infty}\sqrt{\frac{D\tau}{\pi}}\left(\sum_{i=0}^{j-1}\frac{c_s(t_i) - c_{eq}}{\sqrt{j+1-i}} + 2(c_s(t_j) - c_{eq})\right),$$

where

$$c_s(t_i) = \frac{1}{K}\frac{\theta(t_i)}{1-\theta(t_i)}\exp\left(\frac{\theta(t_i)}{1-\theta(t_i)} - \beta\theta(t_i)\right), \quad 0 \le i \le j, \quad j \ge 0.$$

is expressed by using the Van der Waals equation for the adsorption isotherm.

Equivalent Fractional Order ODE. It is well known [1], that the integral equation

$$y(t) = \sum_{\nu=0}^{\lceil\alpha\rceil-1} y^{(\nu)}\frac{t^\nu}{\nu!} + \frac{1}{\Gamma(\alpha)}\int_0^t (t-u)^{\alpha-1} f(u, y(u))du \tag{11}$$

is equivalent to the initial value problem for the fractional order ODE

$$D_*^\alpha y(t) = f(t, y(t))$$
$$y^{(k)}(0) = y_0^{(k)}, \qquad k = 0, 1, \ldots, \lceil\alpha\rceil - 1,$$

where $\lceil\alpha\rceil$ is the smallest integer $\ge \alpha$.

For the Ward and Torday integral equation we have $\alpha = 1/2, \quad \lceil\alpha\rceil = 1$ and the integral equation (11) reads:

$$y(t) = y_0 + \frac{1}{\sqrt{\pi}}\int_0^t \frac{f(u, y(u))}{\sqrt{t-u}}du, \tag{12}$$

where

$$f(u, y(u)) = \frac{\sqrt{D}}{\Gamma_\infty}(c_{eq} - c_s(u)).$$

The equivalent differential problem is

$$D_*^{1/2} y(t) = f(t, y(t)) \tag{13}$$

$$y(0) = y_0. \tag{14}$$

To solve the problem (13)–(14) we use a **modification of the Adams method for fractional order ODEs**, proposed and investigated in [1]. The method is of predictor-corrector type. Applied to the problem (13), (14), it reads:

Predictor scheme:

$$y_{k+1}^P = y_0 + \frac{1}{\Gamma(\alpha)} \sum_{j=0}^{k} b_{j,k+1} f(t_j, y_j),$$

where the coefficients $b_{j,k+1}$ are given by

$$b_{j,k+1} = \frac{h^\alpha}{\alpha}((k+1-j)^\alpha - (k-j)^\alpha).$$

Corrector scheme:

$$y_{k+1} = y_0 + \frac{1}{\Gamma(\alpha)} \left(\sum_{j=0}^{k} a_{j,k+1} f(t_j, y_j) + a_{k+1,k+1} f(t_{k+1}, y_{k+1}^P) \right),$$

where

$$a_{j,k+1} = \frac{h^\alpha}{\alpha(\alpha+1)} \begin{cases} (k^{\alpha+1} - (k-\alpha)(k+1)^\alpha) & j = 0, \\ ((k-j+2)^{\alpha+1} - (k-j)^{\alpha+1} & \\ -2(k-j+1)^{\alpha+1}) & 1 \le j \le k, \\ 1, & j = k+1 \end{cases}$$

Parametric Identification

In the section "Numerical Experiments", we give examples of using two implemented in MATLAB and R functions for numerical optimization. Here we propose a basic iterative algorithm that explains how we can obtain an estimation for the two parameters.

Let us denote

$$\varepsilon(\Gamma_\infty, K) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{\hat{\sigma}^i - \sigma^i}{\hat{\sigma}^i} \right)^2}.$$

where $\hat{\sigma}^i$ are the values of the numerical solution while σ^i are the experimental data.

The algorithm is the following:

1. Begin with an initial estimation for the parameters—$\Gamma_\infty^{(0)}$, $K^{(0)}$.

2. Let us have $(\Gamma_\infty^{(k)}, K^{(k)})$. We solve the differential problem with those values for the parameters to obtain $\varepsilon(\Gamma_\infty^{(k)}, K^{(k)})$.

3. Solve the differential problem with values for the parameters, consecutively, $(\Gamma_\infty^{(k)}+\delta, K^{(k)})$, $(\Gamma_\infty^{(k)}-\delta, K^{(k)})$, $(\Gamma_\infty^{(k)}, K^{(k)}+\delta)$, and $(\Gamma_\infty^{(k)}, K^{(k)}-\delta)$ to obtain an approximation of $\frac{\partial\varepsilon}{\partial\Gamma_\infty}(\Gamma_\infty^{(k)}, K^{(k)})$ and $\frac{\partial\varepsilon}{\partial K}(\Gamma_\infty^{(k)}, K^{(k)})$.

4. Obtain the next estimation as

$$(\Gamma_\infty^{(k+1)}, K^{(k+1)}) = (\Gamma_\infty^{(k)}, K^{(k)}) - \left(\mu\frac{\partial\varepsilon}{\partial\Gamma_\infty}, \nu\frac{\partial\varepsilon}{\partial K}\right),$$

   where μ and ν are determined adaptively, so that the error decreases.

Remark: For the initial estimation of the parameters Γ_∞ and K we propose the following approach.

Starting with an initial value for Γ_∞ and using the surface tension σ at $t = 64$ from the experimental data, we derive the following cubic equation for θ:

$$\beta\theta^3 - \beta\theta^2 + (2 + 2A)\theta - 2A = 0,$$

where

$$A = \frac{\sigma_0 - \sigma(64)}{E_B\Gamma_\infty}.$$

Let us denote the real root of the above equation as θ_1. Substituting θ_1 in the Van der Waals equation for the adsorption isotherm we obtain an initial value for K:

$$K_0 = \frac{1}{c_{eq}}\frac{\theta_1}{1-\theta_1}\exp\left(\frac{\theta_1}{1-\theta_1} - \beta\theta_1\right) \tag{15}$$

Numerical Experiments

First, we compare the computational times for solving the differential problem (1)–(6) and its equivalent formulations by using the different numerical methods. In Table 3, we present computational times for the aforementioned numerical methods. The programs, used for the tests, were implemented in the FORTRAN programming language.

Table 3: Computational times for solving the differential problem with different numerical methods

Numerical method	Computational time
Explicit difference scheme–1	5.48433s
Explicit difference scheme–2	2.04687s
Ward and Torday integral equation–1	1.92554s
Fractional order ODE	1.59375s

Second, we compare the accuracy of the methods for solving the Ward and Torday integral equation on a particular case of this equation:

$$u(t) = 1 - \frac{1}{\sqrt{\pi}} \int_0^t \frac{u(\tau)}{\sqrt{t-\tau}} \mathrm{d}\tau,$$

whose exact solution is known:

$$u(t) = \exp(t)\mathrm{erfc}(\sqrt{t}),$$

erfc being the Complimentary Error Function. As expected, the Adams method for fractional order ODE is more accurate than the modification of the Left Rectangle Rule.

Now, we give some results for the estimated model parameters Γ_∞ and K. Using the explicit difference scheme–1 and the MATLAB procedure "lsqnonlin", we obtain the following values—$\Gamma_\infty = 5.0741 \times 10^{-6}$ and $K = 19.3733$. For those values we obtain the result for σ, that is shown on Figure 3.

Using the algorithm described in the section "Paramametric Identification" and again the explicit difference scheme–1, we obtain similar results—$\Gamma_\infty = 4.9728 \times 10^{-6}$, $K = 20.0028$.

We have used also the modification of the Left Rectangle Rule for the integral equation and a general-purpose optimization R function 'optim' [4] with default options to find the two unknown parameters K and Γ_∞. From previous experiments it is known that Γ_∞ is approximately of order 10^{-6}. We set initial value of $\Gamma_\infty = 0.5 \times 10^{-6}$ and derive K_0 from equation (15). These are our starting values in the optimization procedure. We minimize the relative error equal to $\varepsilon(\Gamma_\infty, K)$, defined above, where $\hat{\sigma}^i$ are the values of the numerical solution while σ^i are the experimental data.

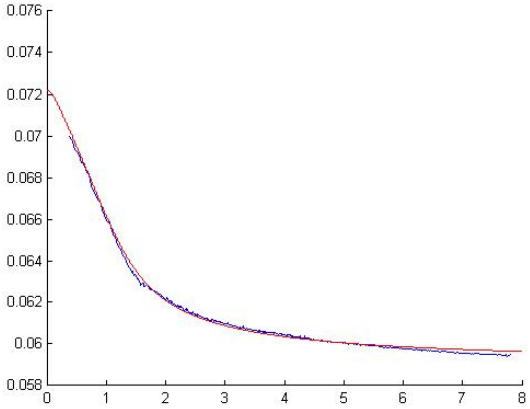


Figure 3: Numerical solution $\sigma(t)$ for model parameters $\Gamma_\infty = 5.0741 \times 10^{-6}$ and $K = 19.3733$

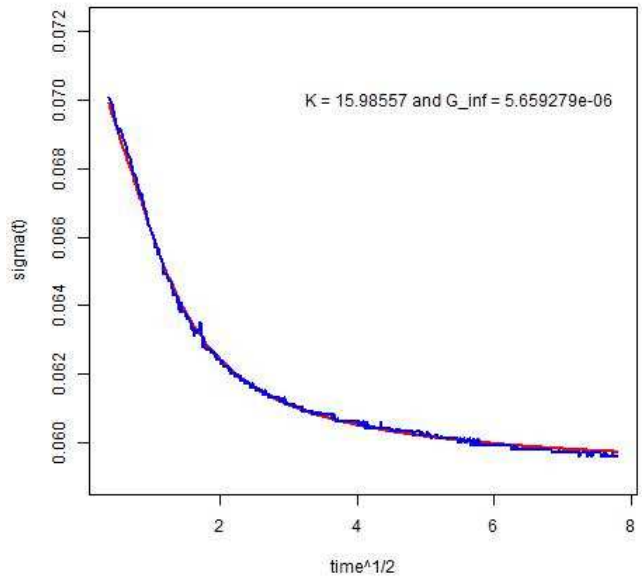


Figure 4: Numerical solution $\sigma(t)$ for model parameters $\Gamma_\infty = 5.659 \times 10^{-6}$ and $K = 15.659$

Conclusion

As a result of the work of our group we present different ways for solving the differential problem (1)–(6) and its equivalent formulations. All of them give similar results, but the Adams method for solving the equivalent fractional order ODE is the fastest one. In addition, this method has better accuracy than the modification of the Left Rectangle Rule for solving the equivalent integral equation.

We also propose different ways for estimating the two unknown parameters—by means of already implemented in MATLAB and R functions, as well as by an algorithm we have implemented.

References

[1] K. Diethelm, N. Ford, A. Freed. Detailed error analysis for a fractional Adams method. Numerical algorithms 36 (1) (2004), 31–52.

[2] P. Englezos, N. Kalogerakis. Applied Parameter Estimation for Chemical Engineers. Marcel Dekker, Inc., 2001.

[3] K. Schittkowski. Numerical Data Fitting in Dynamical Systems. A Practical Introduction with Applications and Software. Springer, 2002.

[4] R Core Team (2014) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Smoothing of well rates in subsurface hydrocarbon reservoir simulators

Ognyan Kounchev, Michail Todorov, Daniela Georgieva,
Nikola Simeonov, Vassil Kolev

1. Introduction

A common problem in reservoir simulators is the history matching problem, where a number of wells are operated at a prescribed flow rate, measured by the operator. The data provides input to a simulator which then has to match various other measured quantities, such as pressure drop at wells, movement of saturation fronts, water break-out and other. A common problem is that the input data is very rough and if input directly would cause considerable numerical difficulties, such as excessive Newton iterations to converge or excessively small time-steps.

2. Posing the Problem

A typical input for a well is a flow rate, specified at discrete time instances, which is positive at every instance. The goal is to replace the "rough" flow rate with a smoother function, which retains two properties of the original:

- It remains positive at every instance;
- The integral over the entire time range is preserved.

Different smoothing scenarios are expected to be seen.

2.1. First Scenario: Approximation by Smoothing Splines and Newton-Raphson Method

Replace the data function $f(t)$ by a smoothing spline $S_f \in C^2$ with restrictions

$$\int_0^T f(t)dt = \int_0^T S_f dt \qquad \text{and} \qquad S_f > 0$$

for $0 \leq t \leq T$.

Let us assume that the data sites $\{t_j\}_{j=1}^N$ with $t_1 < t_2 < \cdots < t_N$ are given with some data $f_j \geq 0$, which are assumed to be the values of a function $f(t)$, i.e.

$$f(t_j) = f_j \qquad \text{for } j = 1, 2, \ldots, N.$$

The problem is to "smoothen" those data f_j, which represent a very abrupt jump, by finding a function $g(t)$, for which the following conditions hold

$$\int_{t_1}^{t_N} L_f(t)\, dt = \int_{t_1}^{t_N} g(t)\, dt$$

$$g(t) \geq 0 \qquad \text{for } t_1 \leq t \leq t_N;$$

here the function $L_f(t)$ is the linear interpolating spline, which satisfies

$$L_f(t_j) = f_j \qquad \text{for } j = 1, 2, \ldots, N.$$

For solving this problem we propose to use approximation (smoothing) cubic splines $S_f(t)$, which by definition belong to $C^2(t_1, t_N)$ [1, 2] (i.e., have two continuous derivatives), having a parameter λ, which provides a trade off between the "goodness of interpolation to the data f_j" and coarseness of the graph of the spline function $S_f(t)$. Such a spline S_f is defined as the **unique solution** of the following problem:

$$\min_{S_f} \left(\lambda \sum_{j=1}^{N} w_j \left(S(t_j) - f_j \right)^2 + (1-\lambda) \int_{t_1}^{t_N} \varphi(t) \left| S''(t) \right|^2 dt \right). \tag{1}$$

Here the parameter λ, the so-called smoothing parameter is given. We assume also: given weights $w_j \geq 0$, which show how good we wish to have the size of $|S(t_j) - f_j|$ for every j, and also the function $\varphi(t) \geq 0$ in the interval $[t_1, t_N]$, which shows the "roughness" of the function $S_f(t)$ at every particular point t. We will not use this large freedom but we will choose $\varphi(t) = 1$.

However we will use essentially the weights w_j in order to satisfy the condition

$$C := \int_{t_1}^{t_N} S_f(t)\, dt = \int_{t_1}^{t_N} L_f(t)\, dt. \tag{2}$$

2.2. Second Scenario: Replace the initial piece-wise linear "rough" curve by another piece-wise linear less "rough" curve

The new curve must be subject to the above restrictions. This is possible to implement and to get a simple explicit relationship to some new average value (see Figure 1):

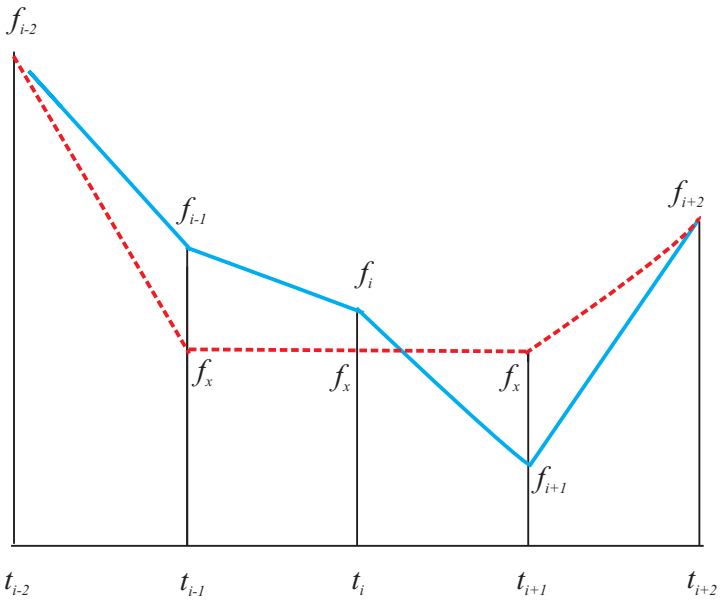


Figure 1: Graph sketch of the moving average. The areas under the solid line (four trapezoidals) and dashed line (two lateral trapezoidals and two congruent rectangles) are equal.

$$f_x = \frac{f_{i-1}(h_{i-1} + h_i) + f_i(h_i + h_{i+1}) + f_{i+1}(h_{i+1} + h_{i+2})}{h_{i-1} + 2(h_i + h_{i+1}) + h_{i+2}} \tag{3}$$

where $h_i = t_i - t_{i-1}$, $i = 1, N$ procedure for $i = 1, \ldots, N$ by step 4. Let us note that it is possible to divide to groups of more trapezoidals but then one can lose the general trend of the original empirical curve. Also, obviously $f_x > 0$ for any empirical data. Applying this procedure one cuts the largest deviations of the measurements.

2.3. Other Scenarios: Approximation by Discrete Wavelet Transform

One can approximate any piecewise continuous function by using a pair of orthogonal bases containing scaling and wavelet functions $\varphi(t)$ and $\psi(t)$. The

scaling function $\varphi(t) = \sum_n \sqrt{2}h(n)\varphi(2t-n)$, where $h(n)$ is the lowpass filter coefficient estimated with $h(n) = \frac{1}{\sqrt{2}}\langle\varphi(t/2), \varphi(t-k)\rangle$, while the wavelet function $\psi(t) = \sum_n \sqrt{2}g(n)\psi(2t-n)$, where $g(n)$ is the highpass filter coefficient. Both filter coefficients are coupled with the explicit relationship $g(n) = (-1)^n h(1-n)$ (see, for example [3] and [4]). The wavelets expand the signals into separate frequency components, and then one can study each component with a resolution matched to its scale. The discrete wavelet transform (DWT) is a special case of wavelet transform that provides a compact representation of a signal in time and frequency that can be computed efficiently. The decomposition of a given function $f(t)$ for j-level by the above basis functions is:

$$f(t) = \sum_n h(n)\varphi(t-n) + \sum_j \sum_n g_{2^j+n}(n)\psi(2^j t - n) = f_a(t) + \sum_j f_{d_j}(t). \quad (4)$$

The first term $f_a(t)$ is the approximation function, while the second term is a sum of the so-called detail functions. An example of decomposition for level $j = 3$ with the orthogonal 10-taps Daubechies wavelet is shown on Figure 2.

Two approaches are possible to be considered:

a) For *uniform mesh (UM)* on the time segment $[0, T]$ the integral of the function $f(t)$ can be approximated for a given j-level by the discrete wavelet decomposition $f_a(t)$ with error ε_0:

$$\int_0^T f(t) = \int_0^T f_a(t) + \varepsilon_0;$$

b) For *non-uniform mesh (NUM)* on the time segment $[0, T]$ the integral of the function $f(t)$ can be approximated also with $f_a(t)$ but with another error ε_1:

$$\int_0^T f(t) = \int_0^T f_a(t) + \varepsilon_1.$$

Simpson's integration rule [5] for smooth functions is preferable compared to the trapezoidal rule [5, 6] since the error is roughly proportional to 10^{-4} and it does not require a dense mesh to attain a priori desired accuracy. Although the trapezoidal rule is inefficient in general, it can be shockingly efficient for very jagged and periodic functions fast approaching zero. This simplest numerical integration technique can be extraordinarily efficient when it is skilfully applied for getting reliable approximations of empirical data and relationships.

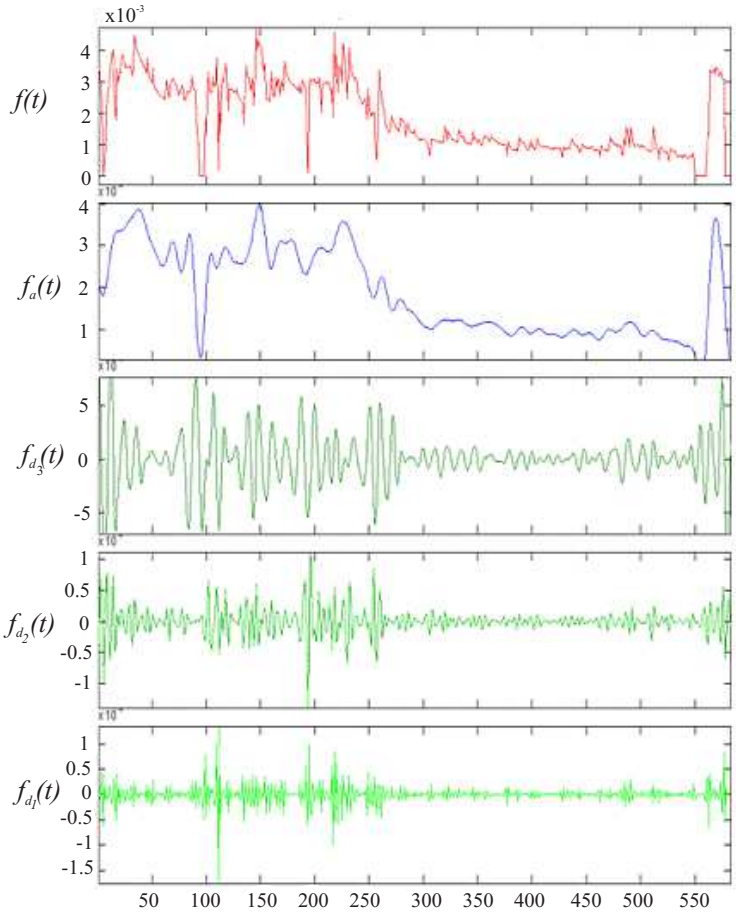


Figure 2: DWT of the customer function for level $j = 3$ with the orthogonal 'db10'

3. Numerical Results

In the particular set of data we have $N = 585$ and the area under the empirical curve is given by the integral

$$C = \int_{t_1}^{t_N} S_f(t)\, dt = 1.316303598725274.$$

For solving the system (1) and (2) we need one more free parameter: we

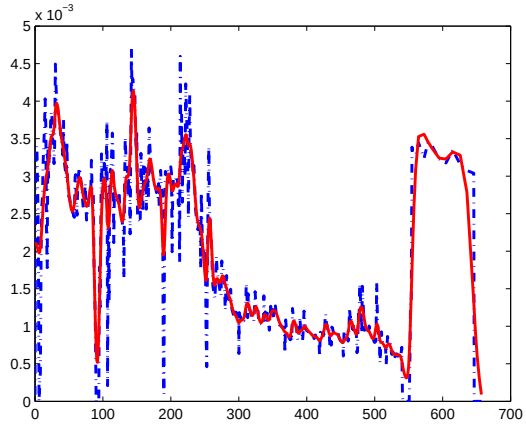


Figure 3: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.08$ (solid line).

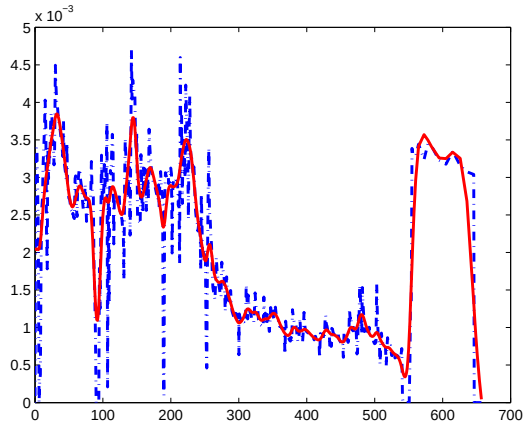


Figure 4: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.01$ (solid line).

consider the unknown weights (with unknown parameter x):

$$w_1 = \cdots = w_{10} = w_{576} = \cdots = w_{585} = x$$
$$w_j = 1 \qquad \text{for } 11 \leq j \leq 575$$

and we solve the system by using a Newton-Raphson solver.

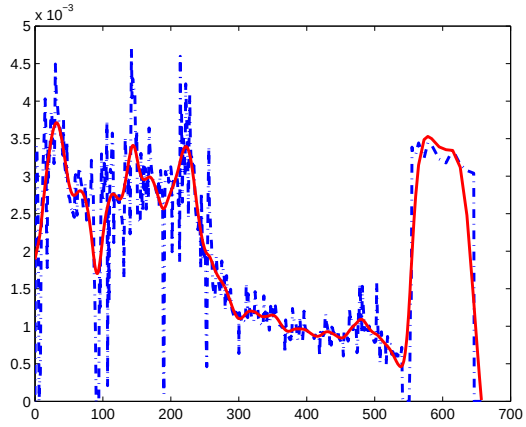


Figure 5: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.001$ (solid line).

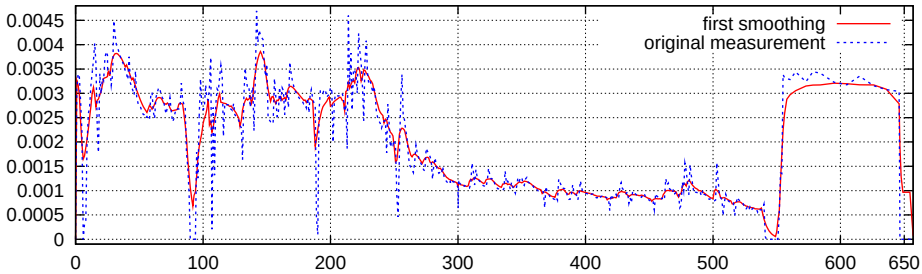


Figure 6: Second scenario: original empirical data (dashed line), piece-wise linear curve with restriction from Figure 1 (solid line).

For different values of the parameter λ we obtain solutions $S_f(t)$, which satisfy condition (2) with precision 10^{-16}.

To demonstrate how does the procedure work we provide some experimental results with different values of the smoothing parameter λ. It is clearly seen on Figures 3, 4, 5 that the smaller values of the parameter λ smooth more the spline $S_f(t)$. Also, condition $S_f(t) \geq 0$ is satisfied. The latter is attained by manipulation of the weights w_j and the function $\varphi(t)$.

For the first scenario – smoothing splines with restrictions and sequential Newton-Raphson technique we get the results: $OldArea = 1.316303598725274$ and $NewArea = 1.316303598725274$. They are practically identical because the

error reaches the machine epsilon, i.e., $Err \sim 10^{-17}$

For the second scenario based on the explicit formula (3) – replacement of one piece-wise linear curve by another one the numerical results cover fully the prediction given by the first scenario of smoothing: $OldArea = 1.31630359872527$, $NewArea = 1.31630359872527$, with $Er \sim 10^{-17}$ (Figure 6).

Since the customer data form a jagged function with fast approaching zero parts the trapezoidal rule for integration in the third scenario is used. The calculated integral value is

$$\int_0^T f(t) = 1.311518426300291 \quad \text{with} \quad T = 585.$$

The minimal-approximation absolute errors for *UM* and *NUM* are tabulated in Tables I and II. Obviously, the accuracy of the approximating integrals depend on the uniform mesh and the levels of DWT. From the level decompositions for *UM* when $j = 1, \ldots, 5$ and *NUM* of $j = 1, 2, 3$ we conclude that the increase of j-level leads to both a decrease of the accuracy, and an increase of the approximation errors ε_0 and ε_1 (see Figures 7 and 8). The higher levels of DWT, however, provide smoother functions, which is the customer preference. The magnitudes of errors of the both methods vary as follows: $\varepsilon_0 \in (10^{-6}, 10^{-4})$, $\varepsilon_1 \in (10^{-4}, 10^{-2})$.

Table I: The minimal approximation errors for *UM*

level	wavelet	$\lvert\varepsilon_0\rvert * 10^{-4}$
1	'sym2'	0.025
2	'db15'	0.890
3	'db2'	3.941
4	'bior3.1'	0.243
5	'db41'	1.105

Table II: The minimal approximation errors for *NUM*

level	wavelet	$\lvert\varepsilon_1\rvert * 10^{-4}$
1	'db42'	0.4668
2	'sym9'	6.9017
3	'db42'	35.44

Conclusion

Which scenario to choose? Actually the developed scenarios are equivalent as a prediction and an order of approximation of the quadratures. Their advantage consists in the fast computer realization provided the output results as input data for the further computer processing and simulations. These procedures are not unique – they can be varied depending on the needs of the user.

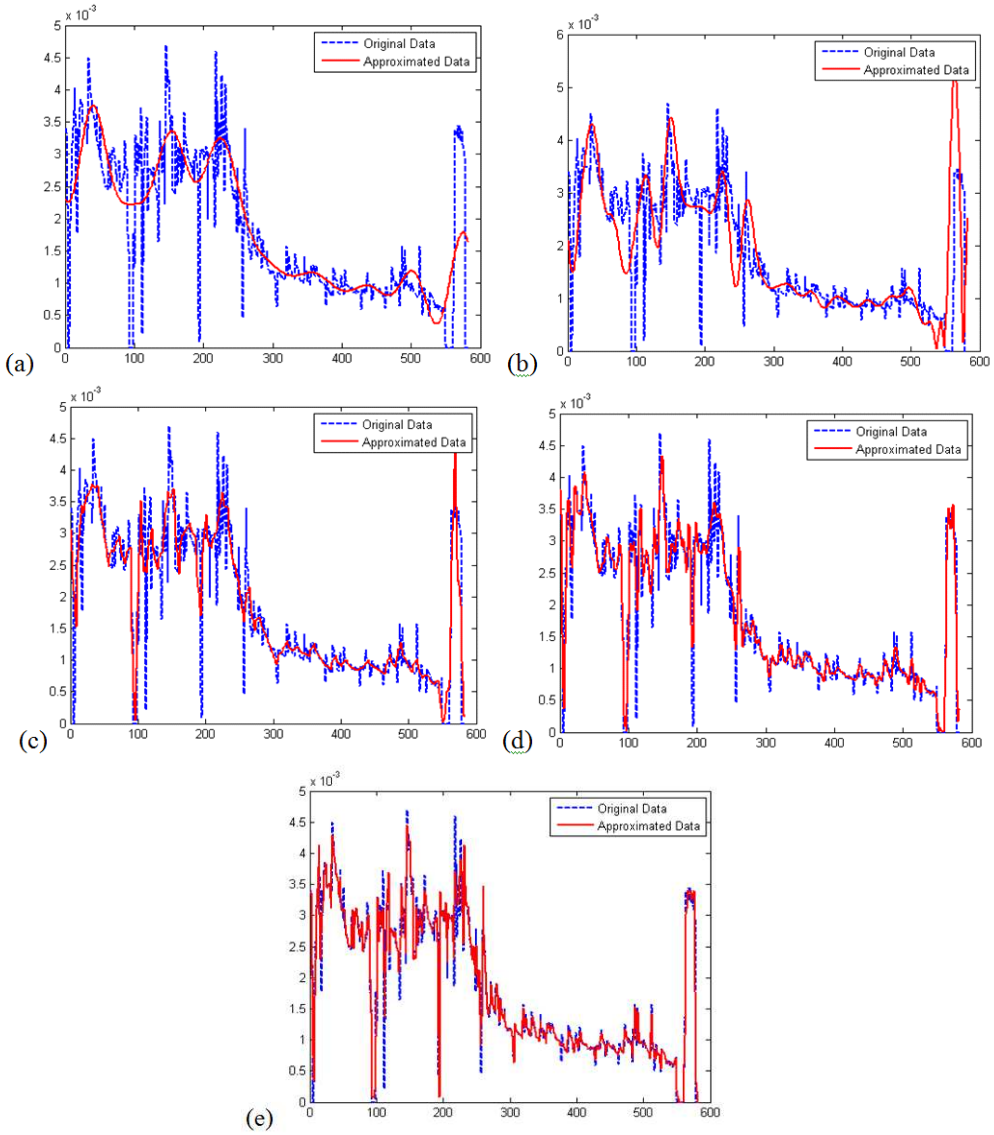


Figure 7: The approximation function for UM with ε_0 for: (a) $j = 5$ with the orthogonal wavelet 'db41'; (b) $j = 4$ with the biorthogonal wavelet 'bior3.1'; (c) $j = 3$ with the orthogonal wavelet 'db2'; (d) $j = 2$ with the orthogonal wavelet 'db15'; (e) $j = 2$ with the symmlet 'sym15'. Empirical data (dashed line), approximation function (solid line)

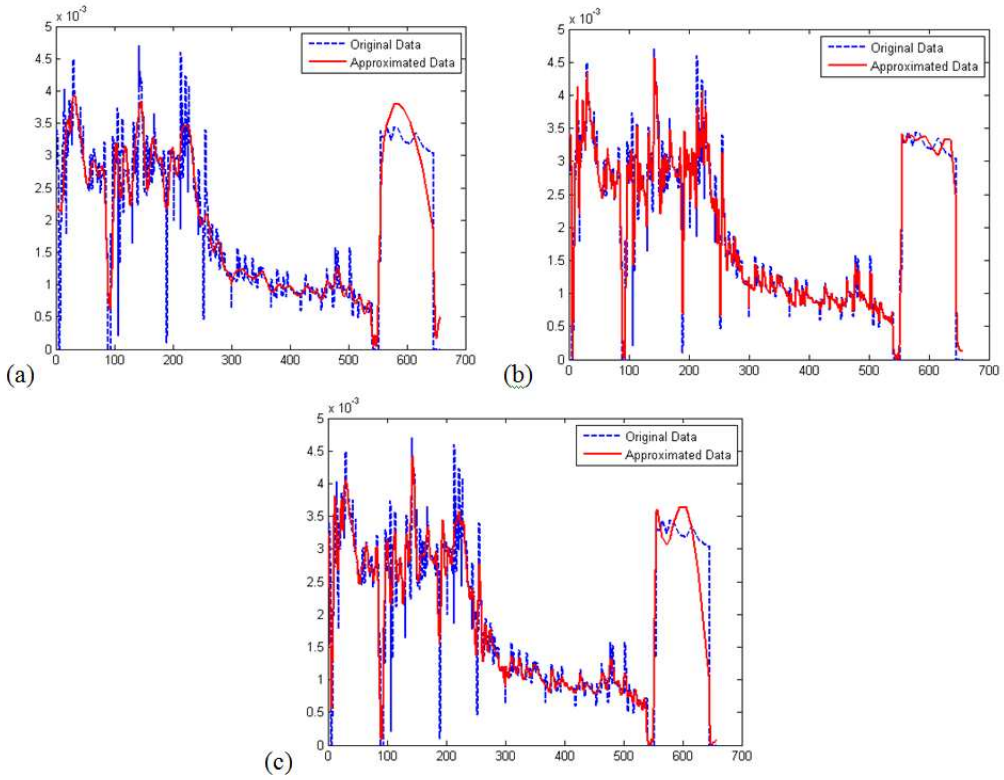


Figure 8: The approximation function for *NUM* with ε_1 for: (a) $j = 3$ with the orthogonal wavelet 'db42'; (b) $j = 2$ with the symmlet 'sym9'; $j = 1$ with the orthogonal wavelet 'db42'. Empirical data (dashed line), approximation function (solid line)

References

[1] C. De Boor, *A Practical Guide to Splines*, Revised Edn., Springer, 2001, pp. 207–214.

[2] R. E. Smith Jr., J. M. Price and L. M. Howser, "A Smoothing Algorithm Using Cubic Spline Functions", NASA TN D-7397, Feb. 1974, Retrieved May 2011.

[3] K. Shukla, A. Tiwari, Efficient Algorithms for Discrete Wavelet Transform With Applications to Denoising and Fuzzy Inference Systems, Springer Briefs in Computer Science, 2013.

[4] O. Christensen and K. Christensen, Approximation Theory: From Taylor Polynomials to Wavelets, Birkhauser Verlag AG, 2004.

[5] J. Epperson, An Introduction to Numerical Methods and Analysis, Willey, 2007.

[6] A. Gilat and V. Subramaniam, Numerical Methods for Engineers and Scientists: An Introduction with Applications using MATLAB, 3rd edn., Willey, 2013.

Finding an effective metric used for bijective S-Box generation by genetic algorithms

Tsonka Baicheva, Dusan Bikov, Yuri Borissov, Limonka Lazarova,
Aleksandra Stojanova, Liliya Stoykova, Stela Zhelezova

Introduction

In cryptography, S-box is a basic component of symmetric key algorithms which performs nonlinear substitution. S-boxes need to be highly nonlinear, so that the cipher can resist linear cryptanalysis.

Let $B = \{0, 1\}$ and $B^n = \{0, 1\}^n$. Every function $f : B^n \to B$ is called *Boolean function* of n variables:

$$B_n = \{f : B^n \to B\},\ |B_n| = 2^{2^n}.$$

Let f_1, f_2, ..., $f_m \in B_n$. Mapping $F : B^n \to B^m$ defined by the rule:

$$F(x) = (f_1(x), f_2(x), \ldots, f_m(x)),$$

is called *vectorial Boolean function* and f_1, f_2, ..., f_m are its coordinate functions.

S-boxes transform n-binary input into m-binary output. Let S be the substitution table of an n-binary input into m-binary output mapping, that is, if $B = \{0, 1\}$,

$$S : B^n \to B^m,\ x = (x_1, x_2, \ldots, x_n) \to y = (y_1, y_2 \ldots, y_m) = S(x)$$

S can be considered as a vectorial Boolean function, consisting of m individual n-variable Boolean functions f_1, f_2, ..., f_m, referred to as coordinate Boolean functions, where $f_k : B^n \to B$ and $f_k(x) = y_k \in B$, $k = 1, 2, \ldots, m$.

The main cryptographic interest has been with reversible, or *bijective*, S-boxes. An $(n \times n)$ S-box S is called bijective, if S is an invertible mapping over B^n. Bijective S-boxes represent permutations of their 2^n inputs.

For cryptographic Boolean functions, $nl(f)$ must be close to the maximum to prevent the system from attacks by linear approximations, correlation attacks, fast correlation attacks.

A Boolean function f on B_2^n is also uniquely determined by its Walsh-Hadamard transform. The Walsh-Hadamard transform f^W of f is an integer valued function defined by:

$$f^W(a) = \sum_{x \in B_2^n} (-1)^{f(x) + \langle a, x \rangle},$$

where $\langle a, x \rangle$ is scalar product.

Linearity $Lin(f)$ of the Boolean function f is defined by using Walsh-Hadamard transform with the following:

$$Lin\,(f) = \max_{a \in B_2^n} |f^W| \geq 2^{n/2}.$$

Linearity and nonlinearity of a Boolean function are connected by the relation:

$$nl\,(f) = 2^{n-1} - \tfrac{1}{2} Lin(f).$$

Walsh-Hadamard Transform spectrum of $f(x)$ is the set of all 2^n spectral coefficients for the elements in B^n.

WHT Spectrum Matrix is the matrix of WHT spectrum of all coordinate Boolean functions.

An S-box S is referred as a *Bent S-box*, if its WHT Spectrum Matrix is entirely flat. Bent S-box has the highest possible nonlinearity. It itself is not suitable for our purposes – it is not balanced and exists only for even $n \geq 2m$. From now on we will talk about bijective S-boxes.

The main criteria for cryptographically strong $(n \times n)$ S-box are:

- High nonlinearity;
- High algebraic degree;
- Balanced structure;
- Good autocorrelation properties.

Our task was to give some suggestions for finding an effective metric used for generation bijective optimal S-Box. Because of the given problem's complexity, our group considered different approaches and we gave a few suggestions for problem solving.

Group suggestions

Bear in mind given problem we focus on achieving good performance according to the nonlinearity criterion finding S-box close to Bent one.

- Change the initial parent pool

  Genetic algorithms represent the heuristic approaches for S-box generation. Each genetic algorithm start with an initial parent pool of bijective S-boxes, P_1, P_2, ..., P_t,. Till now it is used as P_i random or AES S-boxes.

$$\begin{matrix} b_{0,0} & b_{0,1} & & b_{0,2^n-1} \\ b_{1,0} & b_{1,1} & & b_{1,2^n-1} \\ \vdots & \vdots & \dots & \vdots \\ b_{2^n-1,0} & b_{2^n-1,1} & & b_{2^n-1,2^n-1} \end{matrix}$$

Figure 1: S-box – a vectorial Boolean function

$$\begin{matrix} w_{0,0} & w_{0,1} & & w_{0,2^n-1} \\ w_{1,0} & w_{1,1} & & w_{1,2^n-1} \\ \vdots & \vdots & \dots & \vdots \\ w_{2^n-1,0} & w_{2^n-1,1} & & w_{2^n-1,2^n-1} \end{matrix}$$

Figure 2: WHT Spectrum Matrix of S

We propose exponential S-boxes as the initial parent pool. Exponential S-boxes are proven to have good cryptographic properties [1].

- Change the cost function

  In genetic algorithms it's necessary to be able to evaluate how *good* a potential solution is relative to other potential solutions. The *fitness function* is responsible for performing this evaluation and returning a *fitness value*, that reflects how optimal the solution is. In the considered algorithm fitness value is based on two functions: fitness – measuring S-box nonlinearity and cost – measuring flatness of WHT Spectrum Matrix, i.e. how close is it to Bent one. For now cost function is evaluated by:

$$\sqrt[p]{\sum_{j=0}^{2^n-1} |w_{i,j} - w_{i,j+1}|^p}$$

  for $p \geq 1$. The lower its value is the better the solution is.

  ◦ The cost function can be computed by using the maximum of the differences between the spectral coefficients of each coordinate function.

  Let Δ_i be the maximal difference of $i^{-\text{th}}$ coordinate:

$$\Delta_i = \{|w_{i,j} - w_{i,j+k}| : j \in (0, 2^n - 1), j + k \leq 2^n - 1\}.$$

  We can calculate the maximal difference for given S-box as:

$$\Delta_S = \max \Delta_i,\ i \in (1, 2^n - 1).$$

WHT Spectrum Matrix of Bent S-box is entirely flat, so $\Delta_{Bent} = 0$. If $\Delta_{S_1} \approx \Delta_{S_2}$ then the second condition can be used. The vectors of maximal differences for two S-boxes (Δ_1, Δ_2,..., Δ_{2^n-1}) of S_1 and (Δ_1, Δ_2,..., Δ_{2^n-1}) of S_2 are considered and the $\Delta_i = 0$ values are counted and the S-box which has more $\Delta_i = 0$ is chosen as a good one.

◦ Another cost function can be computed by using the dispersion of the WHT spectrum of each coordinate function. Statistical dispersion is zero if all the data are the same and increases as the data become more diverse.

Let a_0, a_1,..., a_{2k} be possible values of the WHT spectrum matrix and $p_{i,j}$ be the probability of appearing a_j in the i^{-th} column. Then the mathematical expectation is

$$E(w_i) = \sum_{j=0}^{2k} a_j p_{i,j}.$$

The dispersion of the i^{-th} column of the WHT matrix with respect to the bent WHT Spectrum ($2^{\frac{n}{2}}$) is:

$$D(w_i) = E(w_i^2) - (2^{\frac{n}{2}}) = \sum_{j=0}^{2k} a_j^2 p_{i,j} - 2^n.$$

The dispersion of the S-Box is:

$$D(S) = \frac{1}{2^n - 1} \sum_{i=1}^{2^n-1} D(w_i).$$

Smaller dispersion means flatter spectrum and better S-box.

- Examine smaller S-Boxes

  Natural requirement for 4 bit S-boxes is an optimal resistance against linearity and differential cryptoanalysis.The optimal values for $Lin(S)$ and $Diff(S)$ are known for dimension $n = 4$, but they aren't determined for higher dimension. More precisely, for any bijective mapping $S : B_2^4 \to B_2^4$ we have $Lin(S) \geq 8$ and $Diff(S) \geq 4$.

  Our suggestion is to examine the behavior of the genetic algorithm on 4×4 S-boxes and compare the results with the already known optimal ones [3].

This can give verification of the method and some suggestions for the cost function.

- New approach

  It is considered Quasigroups as a tool for construction of optimal S-boxes.

  Let $(Q, *)$ be a finite binary groupoid, i.e. an algebra with one binary operation $*$ on the non-empty set Q and $a, b \in Q$. A finite binary groupoid $(Q, *)$ is called a quasigroup if for all ordered pairs $(a, b) \in Q$ there exist unique solutions $x, y \in Q$ of the equations $x * a = b$ and $a * y = b$. This implies the cancellation laws for quasigroup i.e. $x * a = x' * a \Rightarrow x = x'$ and $a * y = a * y' \Rightarrow y = y'$.

  Any quasigroup is possible to be presented as a multiplication table known as Cayley table. Removing the topmost row and the leftmost column of the Cayley table of a quasigroup, results in a Latin square.

  Assuming that $(Q, *)$ is a given quasigroup, for a fixed element $l \in Q$, called leader, the transformation $e_l : Q^r \to Q^r$ is as follows:

$$e_l\left(a_0, a_1, \ldots, a_{r-1}\right) = \left(b_0, b_1, \ldots, b_{r-1}\right) \Leftrightarrow \begin{cases} b_0 = l * a_0 \\ b_i = b_{i-1} * a_i, 1 \leq i \leq r-1 \end{cases}.$$

  The representation of finite quasigroups $(Q, *)$, of order n, where $n \geq 2$ and $n = 2d$ as vector valued Boolean functions, can be used. Every Boolean function $f : F_2^m \to F_2$, can be uniquely written in its Algebraic Normal Form (ANF), by which the algebraic degree can be immediately read off. According to their algebraic degree quasigroups can be divided in two classes, class of linear quasigroups and class of non-linear quasigroups. The class of linear quasigroups has a maximal algebraic degree 1, and all other quasigroups (which maximal algebraic degree is bigger than 1) belong to the class of non-linear.

  Our suggestion is to consider quasigroups as a tool for construction of optimal S-boxes. An algorithm for construction of optimal 4×4 S-box already exists [2]. Cryptographically strong $6 \times 4, 8 \times 8$ and other types of S-boxes could be produced by extending the above algorithm. First, the number of rounds and leaders which are necessary to produce Q-S-boxes with the same quality as already known ones, should be obtained and then, should be determined which of them belong to the class of optimal ones regarding to linear and differential characteristics of S-boxes.

Conclusions

S-boxes play a fundamental role for the security of nearly all modern block ciphers. They are basically used to hide the relationship between the plain text and the cipher text. The S-boxes form the only non-linear part of a block cipher. Therefore, S-boxes have to be chosen carefully to make the cipher resistant against all kinds of attacks. In particular there are well studied criteria that a good S-box has to fullfill to make the cipher resistant against diferential, linear and algebraic cryptoanalyses.

An open problem in cryptography is finding an $(n \times n)$ bijective S-box with nonlinearity nl bounded above by $2^{n-1} - 2^{\frac{n}{2}-1}$, where n is even, to prevent the system from attacks by linear approximations, correlation attacks, fast correlation attacks etc. The proposed problem is in close relation with this, so it is also very difficult problem for solving (AES have $n = 8$ and it is not clear that this kind S-box can be optimal in this dimension). We hope our work helps for moving things a little bit forward.

References

[1] S. Agievich, A. Afonenko, Exponential S-boxes, Cryptology ePrint Archive, Report 2004/024 (2004).

[2] D.Gligoroski, H.Mihajloska, Construction of Optimal 4-bit S-boxes by Quasigroups of Order 4, SECURWARE 2012, The Sixth International Conference on Emerging Security Information, Systems and Technologies (2012), 163–168.

[3] G. Leander, A. Poschmann, On the Classification of 4 Bit S-Boxes, In: Arithmetic of Finite Fields, Lecture Notes in Computer Science Volume 4547 (2007), 159–176.

Cyber threats optimization for e-government services

Veselin Politov, Zlatogor Minchev, Pablo Crotti,
Doychin Boyadzhiev, Marusia Bojkova, Plamen Mateev

Problem Definition

A discrete model of e-government (e-gov) services, encompassing: n different components – state bodies (e.g. ministries, agencies etc., engaged after the legal basis regulations), working during m time intervals are used.

One of the key measures that assure the model reliable work is the prevention from cyber attacks that will block the available e-gov services.

In order to achieve business continuity of these services, a certain amount of funding has to be invested. The correct spending of these funds will assure external interventions block or repairing after passed cyber attacks.

Model for Cyber attacks Optimization

Let a matrix $P = (p_{ij})$ be given, noting probability of a cyber attack in the time moment i, $i = 1, 2, \ldots, m$ and service j, $j = 1, 2, \ldots, n$. Another matrix $C = (c_{ij})$ for damages, resulting from a cyber attack in the interval i, $i = 1, 2, \ldots, m$ concerning the service j, $j = 1, 2, \cdots, n$ is also used.

Different cyber attacks prevention is provided with M funding, distributed amongst the e-gov services in the different time moments.

The aim of such funding distribution is to minimize the "overall damage".

The "overall damage" is the sum of multiplied probabilities for cyber attacks and the resulting damages for the different, used in the model, state bodies and periods. The maximal possible "overall damage" Z (excluding preventive investment) is as follows:

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} p_{ij} c_{ij}.$$

The presented model is similar to [1] but is extended with additional "damage reduction function" – $q(x)$, which is defined for non-negative argument values. The following properties are valid for $q(x) : q(0) = 1$, the function is monotonically decreasing and $\lim_{x \to \infty} q(x) = 0$. This creates a specific damage reduction coefficient, if a certain amount of funding x is invested.

The logic behind is as follows: "more prevention investments – less damages from the expected cyber attacks".

Good examples for $q(x)$ are: e^{-x} and $\dfrac{1}{1+x}$. In this way if we invest x_{ij} funds for prevention of the j-th state body in the i-th moment, the resulting damage is decreased in accordance with the investment towards $p_{ij}q(x_{ij})c_{ij}$ and the aggregated one is:

$$z(x_{ij}) = \sum_{i=1}^{m}\sum_{j=1}^{n} q(x_{ij})p_{ij}c_{ij}.$$

Because of practical limitations, a low investments boundary ε ($x_{ij} \geq \varepsilon$) concerning different periods and services directions is used.

Thus, the following optimization task is formulated:

Find the funding investment distribution that provides a prevention of size M and minimize the "aggregated cyber attacks damage":

$$Z(M) = \min z(x_{ij}) = \min \sum_{i=1}^{m}\sum_{j=1}^{n} q(x_{ij})p_{ij}\ c_{ij} \tag{1a}$$

under the following constraints:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \leq M, \qquad x_{ij} \geq \varepsilon, \quad i = 1 \div m,\ j = 1 \div n. \tag{1b}$$

Modification of the Model

The already described problem could be generalized as follows: the overall sum for cyber attacks countering consists of two components: cyber attacks prevention sum – X and cyber attacks repairing sum – U; $M = X + U$. An example for this, considers a part of M to be used for preliminary insurance from possible cyber attacks or repairing activities, following the idea: "more insurance investments for cyber attacks prevention, less repairing ones".

In order to describe the effectiveness of such an investment we use the function $r(u)$, which is analogous to $q(x)$ and is giving the reduction of a certain cyber attack damage, investing the sum u.

The following new model is accomplished:

$$Z(X,U) = \min z(x_{ij}, u_{ij}) = \min \sum_{i=1}^{m}\sum_{j=1}^{n} q(x_{ij})p_{ij}\, r(u_{ij})\, c_{ij} \tag{2a}$$

under the following constraints:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \le X,\ \sum_{i=1}^{m}\sum_{j=1}^{n} u_{ij} \le U,\ x_{ij} \ge \varepsilon,\ u_{ij} \ge 0,\ i = 1 \div m,\ j = 1 \div n. \quad (2b)$$

Numerical experiments

In order to illustrate the presented models four different tasks concerning different e-government services from real projects experts' data (*E-gov portal*, *Portal for cyber security*, *Cloud services*, *Information Systems for Administrations*) have been solved. The total, prevention and repairing investments have been calculated for the years: 2010, 2015, 2020, 2030.

The values of matrix P (probabilities of cyber attacks) and C (damages, resulting from cyber attacks) are defined by STEMO Ltd. experts, taking into account the trends from [1], [2] and for six areas (facets): 1 – "Human Factor", 2 – "Digital Society", 3 – "Governance", 4 – "Economy", 5 – "New Technologies", 6 – "Environment of living".

The overall sum for cyber attacks countering is $M = 28$ units (with cyber attacks prevention sum $X = 21$ and cyber attacks repairing sum $U = 7$).

The low investments boundary is $\varepsilon = 0.2$. Exponent functions for $q(x)$ and $r(u)$ were used.

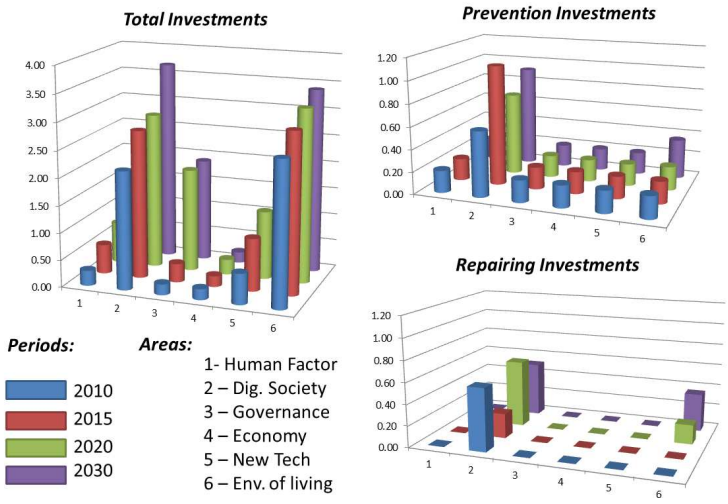


Figure 1: E-gov portal investments for cyber security

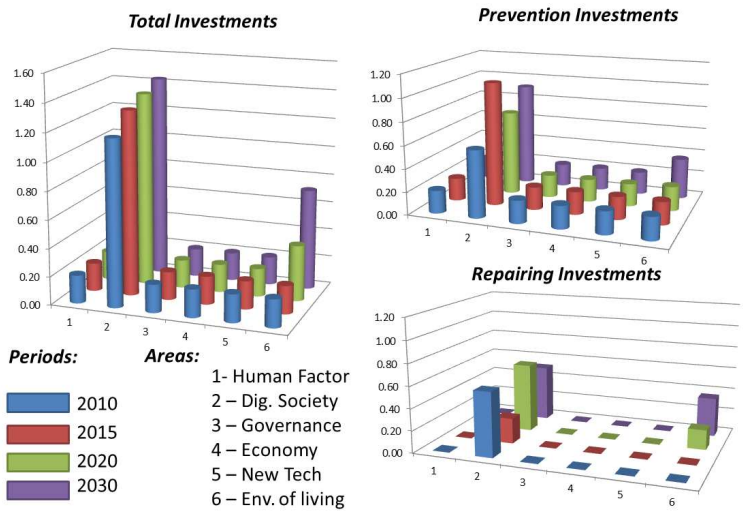


Figure 2: Portal for cyber security investments

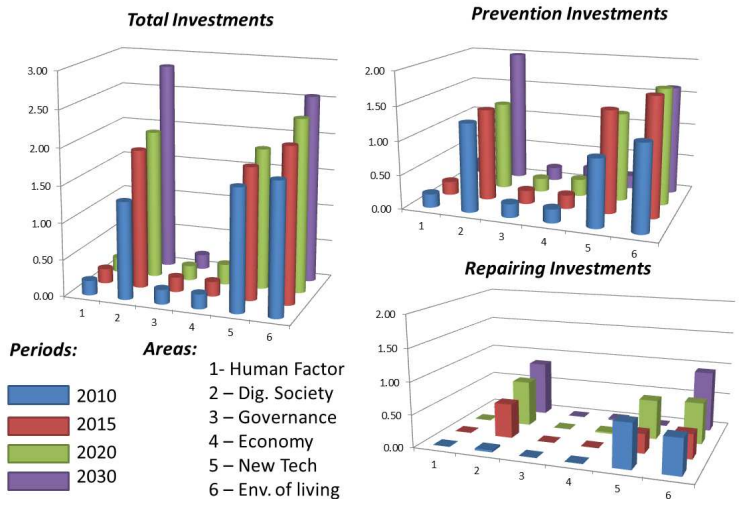


Figure 3: Cloud services cyber security investments

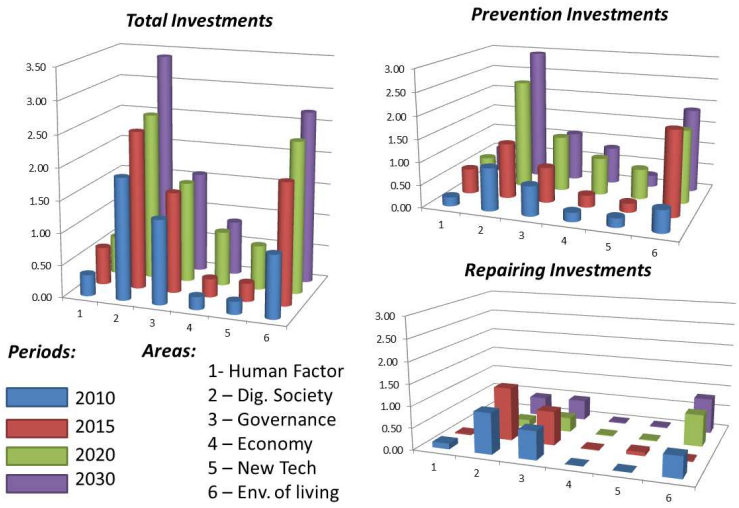


Figure 4: Information Systems for Administrations Cyber Security Investments

As input data could be presented in a small matrix form, MS EXCEL© 2010 product with build-in SOLVER was used [3].

A numerical summary of the results is given in Table 1:

Table 1: Numerical summary of the investments for cyber security

Services/ investments	Total investments	Global losses	Maximum single loss
e-Government Portal	35.00	24.33	1.13
Portal for Cyber Security	10.00	115.09	10.78
Cloud Services	25.00	46.87	2.65
Inf. Systems of Administrations	30.00	33.07	1.50

Discussion

The numerical results obtained from our experimental calculations, though based on some experts' beliefs, are demonstrating a sustainable necessity of growing investments for cyber attacks prevention and repairing for the *Digital society* area, concerning the whole landscape of e-gov services. Apart of this, some other areas like: *New Technologies*, *Environment of living* and *Governance* were also noted as important ones.

The summarized results from Table 1 are outlining also an important point for the implemented model idea regarding the Portal for Cyber Security services, which is with minimal total investments and generates maximum potential global losses.

Obviously, this show the important role of cyber threats prevention investments in general for the created and studied e-gov services in the new digital society.

References

[1] Zlatogor Minchev & Emil Kelevedjiev, Multicriteria Assessment Scale of Future Cyber threats Identification, In: Proceedings of "Mathematics Days in Sofia", July 7–9, 2014, 93–94.

[2] Evangelos Markatos, Davide Balzarotti, Zlatogor Minchev et al., The Red Book – A Roadmap in the area of Systems Security, The SysSec Consortium, 2013.

[3] Christopher Zappe, Ch., Winston W., Ch. Albright, Data Analysis/Optimization/ Simulation Modelling with Microsoft Excel, International Edition, 2011.

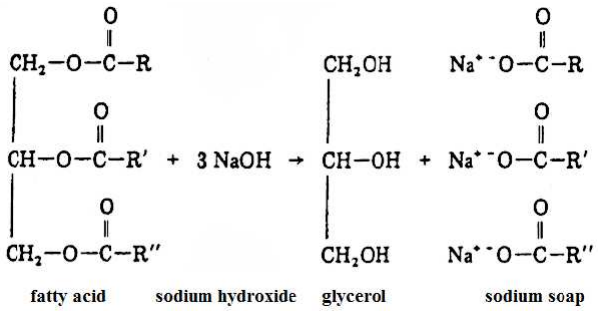
Effect of the precipitation of acid soap and alkanoic acid crystallites on the bulk pH

Gergana Velikova, Ivan Georgiev, Milena Veneva

Introduction

As known from everyday life, physical properties of solutions drastically differ from those of the pure solvents. One distinctive characteristic of the mixtures is their surface tension (γ) after the addition of solute. In many cases, such as multifarious industry productions, the usage of surface-active substances is desired, because of their ability to lower the value of γ. By definition, the surface tension is described as the tension of the liquid molecules on the interface, caused by their interactions with the molecules in the bulk of the liquid, which are thermodinamicaly more favorable.

One possible application of surface-active substances as soaps and detergents are in the emulsion industry. In particular, soaps are widely used for hygienic purposes for hundreds of years. They are products of a chemical reaction between fatty acids (mostly from C_{12} to C_{18} saturated and C_{18} mono-, di- and triunsaturated ones), and sodium or potassium hydroxide in a process called saponification.



Before any of the aforementioned detergents can be used as a cleaning agent they need to be solubilized. In fact, the solubility of soaps and other ionic surfactants depends strongly on the temperature. Therefore, their chemical presence in solution is low, before the temperature reaches the Krafft point. Another key feature of soap colloid solutions is the presence of micelles. As known from the experience, all types of surfactants exist as monomers in the solution, before they reach the critical micelle concentration (CMC), and start self-assembling into micelles. It is important to realize that the soap's micelles cleaning mechanism is

the trapping of substances, which are insoluble in water.

The water solutions of such soaps include different chemical species such as ions of water, soap and hydrogen carbonates, the last ones are results of the solubility of CO_2 in water. Moreover, because of the industry needs, the behavior of the system is examined in the presence of sodium chloride salt and under different acidity.

Formulation of the problem

Chemical settling and goals

One of the most important characteristics of the industrial cleaning products is their optimal pH, which is monitored with addition of fatty acid salts. Given all introductory points, the one component system including sodium soap, water, sodium chloride, and dissolved carbon dioxide is modelled in the paper. Further, chemical species in the complex mixture, coefficients, and constants used in the model formulation are denoted as: K_{A} – fatty acid's dissociation constant; K_{W} – water's dissociation constant; Q_{MZ} – rate constant of the soap production; $K_{\mathrm{CO_2}}$ – used, because of the solubility of the CO_2 from the atmosphere; C_{H} – concentration of the hydrogen cations; C_{Z} – concentration of the fatty acid anions; C_{M} – concentration of the metal cations; C_{MZ} – concentration of the soaps; C_{OH} – concentration of the hydroxide anions; $C_{\mathrm{HCO_3}}$ – concentration of the hydrogencarbonate anions; C_{A} – concentration of the added salt (NaCl); C_{B} – concentration of the added base (NaOH); C_{HZ} – concentration of the undissociated fatty acid. The rate coefficient of soap production and all other dissociation constants are of the type of rate constants. In addition, because of the nature of the manufacturing process, we assume that all reactions are in equilibrium. Therefore, the system of ordinary differential equations from the reaction scheme simplifies to a system of polynomial equations with more than one variable.

$$F_j(x_1, x_2, \ldots, x_N) = b_j, j = \overline{1, N}$$

In that case, we expect to obtain more than one solution and we need to set goals for our numerical implementation of the formulated mathematical model:

goal 1: fast algorithm for solving the system;

goal 2: fast algorithm to detect the positive solution among all of the system's solutions;

goal 3: fitting the theoretically evaluated data for pH with the experimentally obtained one;

goal 4: high precision of the solution.

Mathematical Model

The system of polynomial equations is in the form

$$\begin{aligned} C_{\mathrm{H}} C_{\mathrm{Z}} \gamma_{\pm}^2 &= K_{\mathrm{A}} C_{\mathrm{HZ}} \\ C_{\mathrm{M}} C_{\mathrm{Z}} \gamma_{\pm}^2 &= Q_{\mathrm{MZ}} C_{\mathrm{MZ}} \\ C_{\mathrm{H}} C_{\mathrm{OH}} \gamma_{\pm}^2 &= K_{\mathrm{W}} \\ C_{\mathrm{H}} C_{\mathrm{HCO_3}} \gamma_{\pm}^2 &= K_{\mathrm{CO_2}} \\ I = C_{\mathrm{H}} + C_{\mathrm{M}} &= C_{\mathrm{OH}} + C_{\mathrm{HCO_3}} + C_{\mathrm{Z}} + C_{\mathrm{A}} \\ m_{\mathrm{M}} &= C_{\mathrm{T}} + C_{\mathrm{A}} + C_{\mathrm{B}} - C_{\mathrm{M}} - C_{\mathrm{MZ}} \\ m_{\mathrm{Z}} &= C_{\mathrm{T}} - C_{\mathrm{Z}} - C_{\mathrm{HZ}} - C_{\mathrm{MZ}} \end{aligned} \tag{1}$$

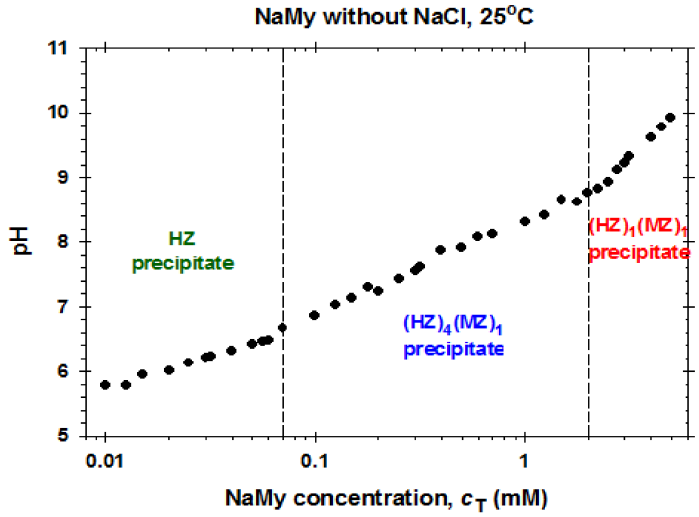
where $K_{\mathrm{W}} = 6.81 \times 10^{-15}$ M^2, $K_{\mathrm{A}} = 1.995 \times 10^{-5}$ M, and $Q_{\mathrm{MZ}} = 2.84$ M. The activity coefficient $\gamma_{\pm}$ is calculated from the semi-empirical formula:

$$\log_{10} \gamma_{\pm} = 0.055 I - \frac{0.5115\sqrt{I}}{1 + 1.316\sqrt{I}} \tag{2}$$

where I is the ionic strength and

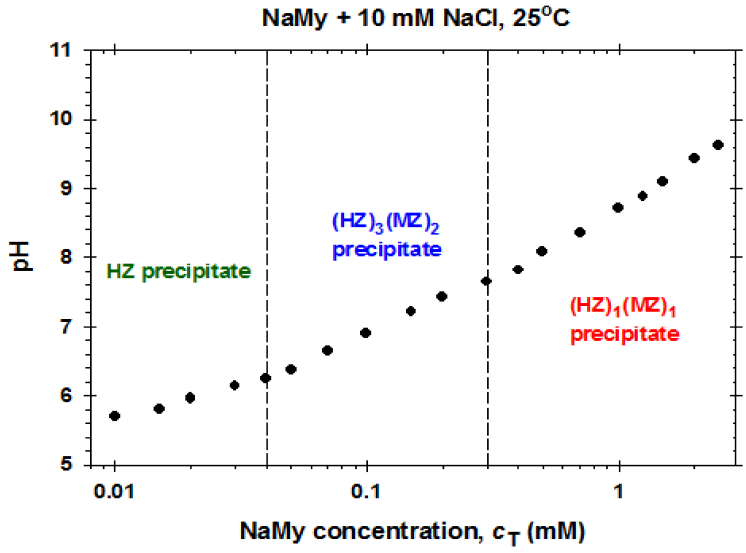
$$\mathrm{pH} = -\log_{10}(\gamma_{\pm} C_{\mathrm{H}}). \tag{3}$$

First case – without NaCl



- $C_{\mathrm{A}} = 0$ M and $C_B = 0$ M

Second case – with NaCl



- $C_{\mathrm{A}} = 0.01$ M and $C_{\mathrm{B}} = 0\,\mathrm{M}$

First and second case – first interval

- solution with fatty acid precipitates
- $C_{\mathrm{HZ}} = S_{\mathrm{HZ}} = 5.25 \times 10^{-7}$ M
- $m_{\mathrm{M}} = 0$

$\Rightarrow$ fit $K_{\mathrm{CO_2}}$
$\Rightarrow$ comparison between the obtained $K_{\mathrm{CO_2}}$ values in the two cases.

First and second case – second interval

- solution with precipitate of $j : n$ acid soap
- $\dfrac{m_{\mathrm{M}}}{n} = \dfrac{m_{\mathrm{Z}}}{n+j}$
- $C_{\mathrm{H}}^{j} C_{\mathrm{M}}^{n} C_{Z}^{j+n} \gamma_{\pm}^{2j+2n} = K_{jn}, \quad \text{if } j = 4 \text{ and } n = 1$
- $C_{\mathrm{H}}^{j} C_{\mathrm{M}}^{n} C_{\mathrm{Z}}^{j+n} \gamma_{\pm}^{2j+2n} = K_{jn}, \quad \text{if } j = 3 \text{ and } n = 2$

$\Rightarrow$ fit K_{41}
$\Rightarrow$ fit K_{32}

First and second case – third interval

- solution with precipitate of $j : n$ acid soap
- $\dfrac{m_{\mathrm{M}}}{n} = \dfrac{m_{\mathrm{Z}}}{n+j}$
- $C_{\mathrm{H}}^{j} C_{\mathrm{M}}^{n} C_{\mathrm{Z}}^{j+n} \gamma_{\pm}^{2j+2n} = K_{jn}, \quad \text{if } j = 1 \text{ and } n = 1$

$\Rightarrow$ fit K_{11}
$\Rightarrow$ comparison between the obtained K_{11} values in the two cases.

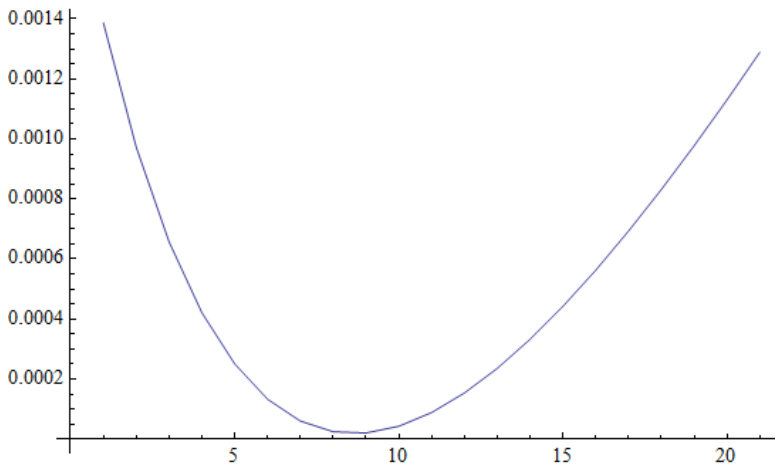
Solution

In order to fit the theoretically evaluated data with the experimentally obtained one, we minimize the following functional:

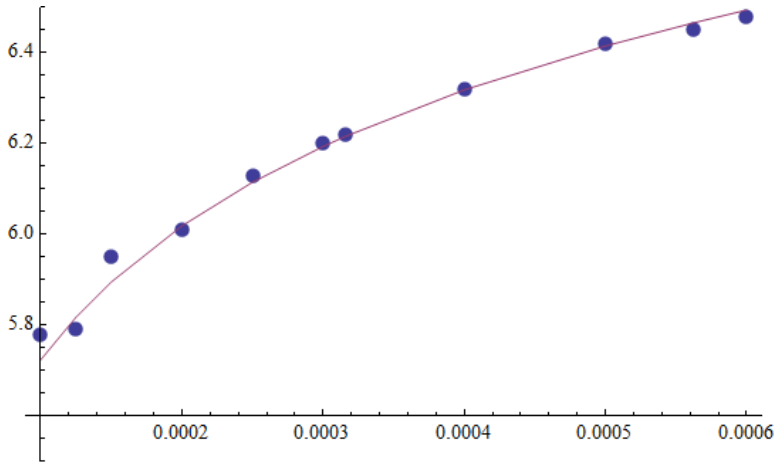
$$P(K_{\mathrm{CO_2}}) = \frac{1}{n} \sum_{k=1}^{n} \left[1 - \frac{\mathrm{pH}_{th}(k)}{\mathrm{pH}_{exp}(k)} \right]^2$$

by numerical variation of $K_{\mathrm{CO_2}}$. Here pH_{th} are the values for pH obtained from (1)–(3) and pH_{exp} are the measured experimental data. Using software for symbolic computations (like Mathematica) one can find a good initial approximation for the parameter $K_{\mathrm{CO_2}}$.

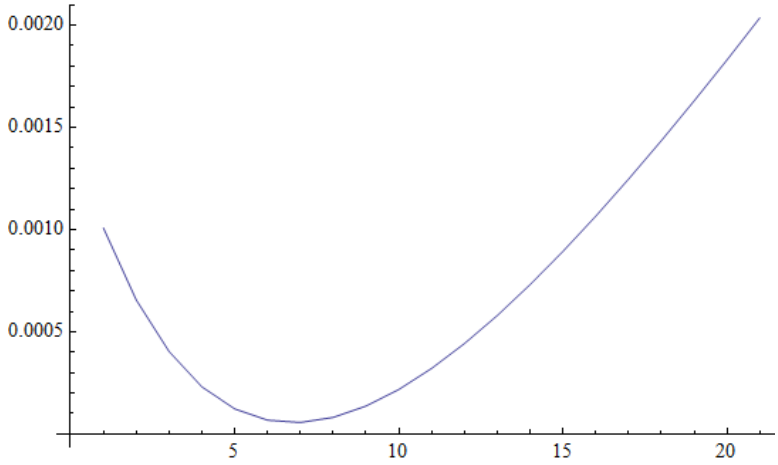
First case (first interval) – values of $P(K_{\mathrm{CO_2}})$**,** $n = 20$



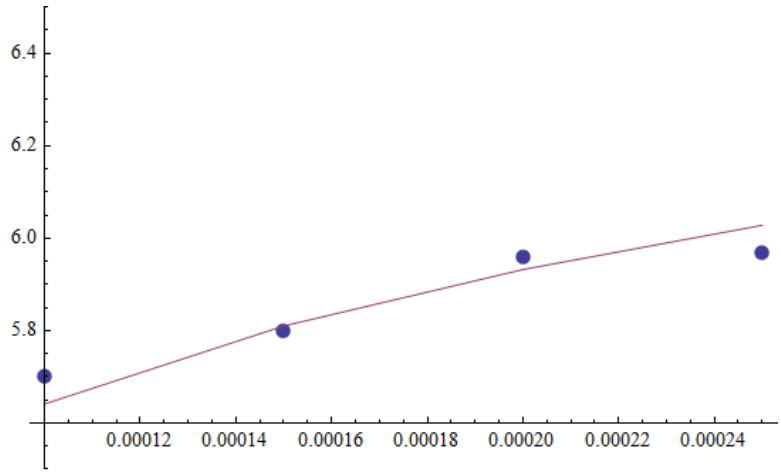
First Case (first interval) – fit of the theoretically evaluated data for pH with the experimentally obtained one ($K_{CO_2} \approx 1.8 \times 10^{-10}$)



Second Case (first interval) – values of $P(K_{CO_2})$, $n = 20$



Second Case (first interval) – fit of the theoretically evaluated data for pH with the experimentally obtained ones ($K_{CO_2} \approx 2 \times 10^{-10}$)



Using the obtained value of K_{CO_2} and the same technique one can fit the parameters K_{32} and K_{11} for the second and respectively the third interval.

Fast algorithm for finding the positive solution

So far we have talked about solving the system of equations we have and fitting the theoretically evaluated data for pH with the experimentally obtained one. However, a very important step of the problem solving is to detect quickly the positive solution aomung the whole set of the system's solutions.

The problem now is the following:

- we have a system of no more than 20 polynomial equations;
- there is no estimation for the number of the solutions that such a system can have, because this number depends on the type of the crystals that are used;
- the components of the solutions could be complex numbers;
- according to a hypothesis from the practice the system can have only one positive solution.

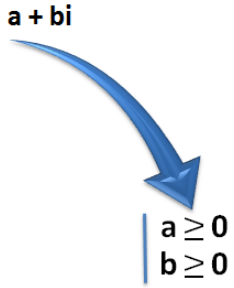
The aim is a fast algorithm to detect the positive solution.

We are going to show two different algorithms, each of them was implemented both in *C++* and *Matlab*. In order to compare the two algorithms, we have been given an example – system, which consists of 16 equations with 16 variables.

The solutions obtained with *Mathematica* are 9, only one of which is positive. For the needs of the computer programs we have written, we assume that each component of each solution is a complex number.

First approach

The first approach is to compare each component of each solution with 0:



So, the algorithm is the following: we take the first component of the first solution. If the real part of this component is not negative, then we compare the imaginary part of this component with 0. If this part is also not negative, we take the second component of the current solution and continue in the same manner. If we find a negative part in a component, we reject the current solution and continue with the next one. Because of the fact that existence of only one positive solution is just a hypothesis, our algorithm does not stop if it finds a solution, which consists of only positive components, but continues searching for other positive solutions.

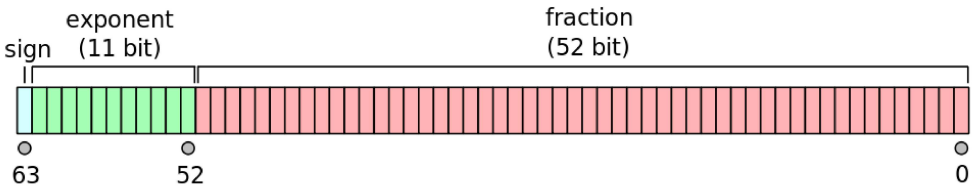
This way, the complexity of the first algorithm is $O(n * m)$, where n is the number of the solutions of the system and m is the number of the components in each solution.

Second approach

In order to garantee the needed precision of the solution, we represent the real and the imaginary part of each component of each solution as a double-precision floating-point number. The benefit is that each double-precision floating-point number has 15 decimal digits in the decimal part of the mantis and the absolute value of such a number is between 10^{-308} and 10^{308}.

Each double-precision floating-point number is represented in the computer's memory as $8B = 64$ bits (according to the standard IEEE). In the picture below you can see what each of these 64 bits is used for. The most important bit for

our second approach is the sign bit. It contains 0 if the number is ≥ 0 and 1, if it is negative.



Thus, the second approach is the following: instead of comparing lexicographically all the bits in the binary representation of a number with the binary representation of 0, as we did in our first approach, we compare only the sign bit of the current number with the sign bit of 0, which is 0. The remaining part of the first algorithm is not changed.

Then:

- the complexity of the algorithm *comparison with 0* is: $O(l * n * m)$;
- the complexity of the algorithm *bit comparison* is: $O(n * m)$,

where l is the number of the bits in the binary representation of the numbers, which we consider. In our case it is 64.

In the worst case scenario, the second algorithm works as fast as the first one. It depends on the optimizations that the processor makes.

Comparison between the two algorithms

C++/Fortran vs. Matlab/Mathematica

- *C++* and *Fortran* are compiled programming languages, which means that the source code of the program is transformed into a machine code before the execution of the program;
- *Matlab* and *Mathematica* are interpreted programming languages, which means that the programs are executed directly, which usually makes them slower because of the overhead of the processor.

$\Rightarrow$ *C++* and *Fortran* are better for scientific computations.

Implementation with MATLAB – time (in seconds)

Bit Comparison	Comparison with 0
3.683144e-005	2.888495e-005
5.576608e-005	3.135687e-005
2.870890e-005	2.804986e-005
4.362872e-005	4.470864e-005
3.317848e-005	3.355881e-005

A number of tests (~ 50) were made. Only two of them show that the algorithm bit comparison is faster than the algorithm comparison with 0 (these are the results in the last two rows at the table below). According to all of the other tests (such results are shown in the first three rows at the table below) we conclude that the algorithm *bit comparison* is slower than the algorithm *comparison with 0*. The reason is that the function, which *Matlab* uses for finding the sign bit, probably has the following implementation (with some optimizations): $\operatorname{sign} v = -(v < 0)$. We cannot be sure, because the function is build-in. The same situation is observed in *Mathematica*. So, using of *Matlab* (and *Mathematica*, too) for solving this problem cannot give us satisfying results.

Implementation with C++ – time

As an example we consider a system having 9 solutions, each with 16 components:

- the average time of the algorithm *comparison with 0*: 1 μs;
- the average time of the algorithm *bit comparison*: 0 μs.

Number of Solutions	Time (µs) - C++		Time (µs) - Matlab	
	Bit Comparison	Comparison with 0	Bit Comparison	Comparison with 0
9	0	1	29	28
801	15	18	560	597
1601	31	39	1090	1113
8001	186	237	5377	5454

This means that the average time of the algorithm *bit comparison* is in nanoseconds. In order to compare the average time for the execution of both implementations of the two algorithms, we test them for bigger number of solutions. In the table above one can see that for 8001 solutions within which only one is positive the algorithm *comparison with* 0 is slower than the algorithm *bit comparison* and the difference in times is 50 μs.

References

[1] Peter Kralchevsky, Krassimir Danov, Cenka Pishmanova, Stefka Kralchevska, Nikolay Christov, Kavssery Ananthapadmanabhan, Alex Lips. *Effect of the Precipitation of Neutral-Soap, Acid-Soap, and Alkanoic Acid Crystallites on the Bulk pH and Surface Tension of Soap Solution.* Langmuir (2007), 23, 3538–3553.

[2] Mariana Boneva, Krassimir Danov, Peter Kralchevsky, Stefka Kralchevska, Kavssery Ananthapadmanabhanb, Alex Lipsc. *Coexistance of micelles and crystallites in solutions of potassium myristate: Soft matter vs. solid matter.* Colloids and Surfaces A: Physicochem. Eng. Aspects 354 (2010) 172–187.

[3] Krassimir Danov, Peter Kralchevsky, Kavssery Ananthapadmanabhan. *Miccele-monomer equilibra in solutions of ionic surfactants and in ionic-nonionic mixtures: A generalized phase separation model.* Advances in Colloid and Interface Science 206 (2014) 17–45.

[4] K. Birdi. *Surface and Colloid Chemistry: Principles and Applications* (2009), 244 pages.

[5] Peter Atkins, Julio de Paula. *Physical Chemistry.* 9th Edition (2009), 972 pages.

[6] Preslav Nakov, Panayot Dobrikov. *Programirane=++Algoritmi.* 3rd Edition (2005), 703 pages.

Circular arc spline approximation of pointwise curves for use in NC programing

Ana Avdzhieva, Dragomir Aleksov, Ivan Hristov, Nikolai Shegunov,
Pencho Marinov

1. Introduction

We consider a numerical control (NC) cutting machine which can cut only line segments and circular arcs. Thermal cutting processes require constant tool velocity because

- too slow velocity leads to overheating and melting,
- too fast velocity interrupts the cutting process.

The inputs with which the machine works are sets of points in a particular order which are in Cartesian plane.
From a set of points (inputs) we must create a sequence of line segments and circular arcs that pass through some of the points and are "sufficiently close" to the others – ϵ error condition. The case in which the points can be approximated with straight line segments is well investigated. We are interested in the sets of points which can only be approximated by arcs. Below we formulate this particular task.

2. The problem

A sequence of N points is given. A curve must be created, composed of circular arcs, such that:

- it passes through/nearby the given points in the same sequence;
- the Hausdorff distance between the points and the curve does not exceed a certain value ϵ;
- it is composed of minimal number of arcs;
- the output should consist of sets of the type:
  $\{(x_1, y_1), (x_2, y_2), (x_c, y_c), E\}$,

where (x_1, y_1) and (x_2, y_2) are respectively the initial and the final points of a certain arc, (x_c, y_c) is its center and $E = +1$ if the direction of the arc is counter

clockwise or $E = -1$ if the direction of the arc is clockwise.

Remark. Local minimum – fitting an arc to each set of 3 points – is not a solution of the task.

2.1. Summary of the approach

- We begin with a program for finding the center and the radius of a circle that passes through three fixed points.
- Having such a program we make another one for finding the "best" arc that connects two fixed points (which have at least two inner points between them). This arc passes through the two fixed points and through one of the points between them.
- Next we find the "best" arc between any two points (that have at least two inner points) of the set of points we are given.
- From the set of arcs that we have created, we exclude those that do not satisfy our error condition.
- From the arcs that are left we may choose different ways to get from the initial point to the last. We chose such a path that contains minimal number of arcs. Usually the connecting points are spread almost uniformly throughout the set we are given.

2.2. An arc through three fixed points

Let us have the points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$, $P_3(x_3, y_3)$, Fig. 1. The midpoints A and B of the line segments connecting (x_1, y_1) and (x_2, y_2) and (x_2, y_2) and (x_3, y_3) have coordinates (x_A, y_A), (x_B, y_B). Obviously

$$x_A = \frac{x_2 + x_1}{2} \ , \ x_B = \frac{x_3 + x_2}{2}$$

and

$$y_A = \frac{y_2 + y_1}{2} \ , \ y_B = \frac{y_3 + y_2}{2}.$$

The equations of the lines that pass through the points $P_1(x_1, y_1)$, $P_2(x_2, y_2)$ and $P_2(x_2, y_2)$, $P_3(x_3, y_3)$ are respectively

$$l_1 : A_1 x + B_1 y + C_1 = 0$$

and

$$l_2 : A_2 x + B_2 y + C_2 = 0,$$

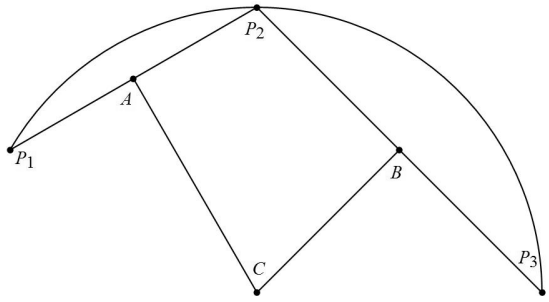


Figure 1: The center C of the circle through P_1, P_2, P_3

where $A_1 = y_2 - y_1$, $B_1 = x_2 - x_1$, $C_1 = -x_1(y_2 - y_1) + y_1(x_2 - x_1)$, $A_2 = y_3 - y_2$, $B_2 = x_3 - x_2$, $C_2 = -x_2(y_3 - y_2) + y_2(x_3 - x_2)$. Now, since the vectors $p_1(A_1, B_1)$ and $p_2(A_2, B_2)$ are orthogonal respectively to the lines l_1 and l_2 and we have the coordinates of A and B, we can easily find the equations of the line bisectors of the arcs that are orthogonal to l_1 and l_2 and pass respectively through (x_A, y_A) and (x_B, y_B). We have

$$b_1 : B_1 x - A_1 y + (-B_1 x_A + A_1 y_A) = 0,$$

$$b_2 : B_2 x - A_2 y + (-B_2 x_B + A_2 y_B) = 0.$$

The center $C(p, q)$ of the circle is where the two line bisectors intersect. Its coordinates are the solution of the system

$$B_1 x - A_1 y + (-B_1 x_A + A_1 y_A) = 0,$$

$$B_2 x - A_2 y + (-B_2 x_B + A_2 y_B) = 0.$$

So we have that

$$p = -\frac{-A_2 B_1 x_A + A_1 B_2 x_B + A_1 A_2 y_A - A_1 A_2 y_B}{A_2 B_1 - A_1 B_2},$$

$$q = -\frac{-B_1 B_2 x_A + B_1 B_2 x_B + A_1 B_2 y_A - A_2 B_1 y_B}{A_2 B_1 - A_1 B_2}$$

As for the radius of the circle, it is equal to the distance between the center and any point on it. We can use the point $P_1(x_1, y_1)$. We have that

$$r = \sqrt{(x_1 - p)^2 + (y_1 - q)^2}.$$

The direction of the arc is positive (negative) exactly when the orientation of the triangle $\overrightarrow{P_1P_2P_3}$ is positive (negative). This orientation is equal to the sign of the determinant

$$\begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{vmatrix}.$$

2.3. "Best" arc

Let us consider the task for connecting two fixed points $P_0(x_0, y_0)$ and $P_{n+1}(x_{n+1}, y_{n+1})$ (which have n inner points, $n \geq 2$) of our input set. First we build all the arcs that connect the two end points and pass through an inner one - that makes n arcs. Let r_i and $C_i(p_i, q_i)$, $i = 1, \ldots, n$ be respectively the radii and the centers of these arcs. For every arc with a center (p_i, q_i) and radius r_i, $(i = 1, \ldots, n)$ we calculate its Hausdorff distance to the inner points P_j, $j = 1, \ldots, n$.

$$d_{i,j} = \left| \sqrt{(x_j - p_i)^2 + (y_j - q_i)^2} - r_i \right|.$$

We now denote

$$d_i := \max\{d_{i,1}, \ldots, d_{i,n}\}.$$

For the i-th arc d_i is its greatest Hausdorff distance to an inner point. We remind that we now consider all the arcs that connect two fixed points and pass through a third between them. For the "best" arc of such kind we chose the k-th arc for which

$$d_k = \min\{d_1, \ldots, d_n\}.$$

"Best" arc – new suggestions.

The input set is the same: two fixed points $P_0(x_0, y_0)$ and $P_{n+1}(x_{n+1}, y_{n+1})$ (which have n inner points, $n \geq 2$). The midpoint M of the segment P_0P_{n+1} has coordinates (x_M, y_M). Obviously

$$x_M = \frac{x_0 + x_{n+1}}{2}, \qquad y_M = \frac{y_0 + y_{n+1}}{2}.$$

The equations of the line that passes through the point M and is perpendicular to the segment P_0P_{n+1} are:

$$c : \begin{cases} x_C = x_M + d * y10/w \\ y_C = y_M + d * x01/w \end{cases}$$

where: $x01 = x_0 - x_{n+1}$, $\quad y10 = y_{n+1} - y_0$, $\quad w^2 = (x01)^2 + (y10)^2$.

For $i = 1, \ldots, n$ we calculate the oriented distance d_i from M to the C_i-center of the circle through the points P_0, P_i, P_{n+1}

$$d_i = \frac{((x_i - x_M)^2 + (y_i - y_M)^2 - w^2/4).w}{2((x_i - x_M).y10 + (y_i - y_M).x01)}, \qquad d = \frac{1}{n}\sum_{i=1}^{n} d_i.$$

Next we define the center C of the optimal arc: C is at distance d from M. The radius of the arc is $r = \sqrt{d^2 + w^2/4}$. We calculate the errors e_i for the points P_i. Note that $e_i = \sqrt{(x_i - x_M)^2 + (y_i - y_M)^2} - r$ is the Euclidean distance between P_i and the point Q_i, which lies on this circle and on the radius through the point P_i. At the same time e_i is the Hausdorff distance between P_i and the optimal arc. More precisely this is one-side Hausdorff distance from given points to the found arc.

2.4. Next stages

Now we consider all the combinations of two points from our input set that have at least two inner points. For all such pairs of points we take the best (according to one of the ways previously described) arc that connects them. Since not all these arcs are close enough to all of their inner points (for an example we can rarely connect the first and last point with only one arc) we exclude those for which the distance between them and their inner points (at least one of them) is more than ϵ. Now we have a set of suitable arcs.
We may consider the problem for constructing a curve (made of arcs) from the first to the last point as a question for finding a path in a graph. We consider each point of the input set as a node and the arcs (connecting some of them and satisfying the error condition) as ribs.

For construction of the adjacency matrix $A = (a_{ij})_{i=1,\ldots,N}^{j=0,\ldots,N}$ we first set A to have only zeros. For $i = 1, \ldots, N-3$ (N is the number of the input points) we consider the best arc (rib) connecting the i-th and the j-th points ($j = i + 3, \ldots, N$). If this arc satisfies the error condition we predefine $a_{ij} = 1$.
We compare different paths by the length of their shortest arc (according to the number of inner points). One approach is to find all the paths in the graph we have derived and then chose the one in which the shortest arc is as long as possible. However, we have adapted an algorithm for finding a path with smallest amount of ribs. Usually the nodes we get are spread uniformly.

3. Numerical experiments

We have applied our approach to real examples. On Figure 2 the black curve consists of 200 points, that lie on the parabolic curve $y = 300 - 200 * (1 - x/500)^2$ and the white inner segments are the arcs (6 is their number), approximate the points.

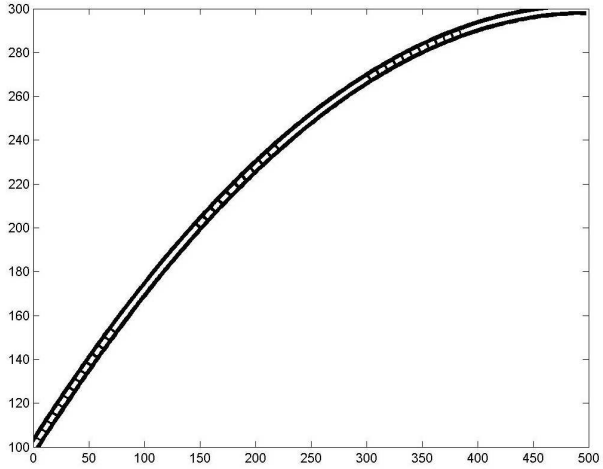


Figure 2: Approximation of the data by 6 arcs

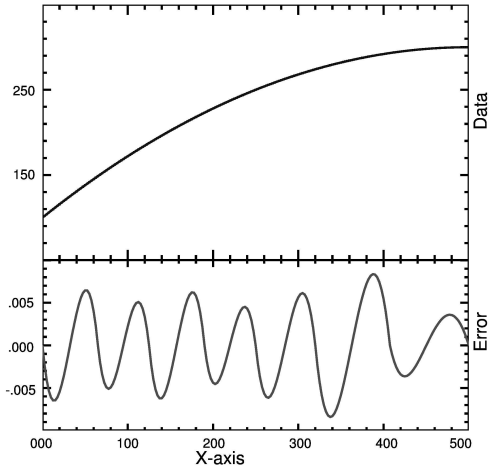


Figure 3: Approximation by 7 arcs (above) and the error of approximation (below)

On Figure 3 we show the approximation of the same data by 7 arcs and below we demonstrate how the error of approximation changes. The maximal error with 5 arcs is about 0.0183, but with 7 arcs – less than 0.0085. The output data for these two cases are:

Number of arcs is $N_{arc} = 5$

```
A (500.000,300.000) (370.000,286.480) (500.58361,  -337.37167)  1
A (370.000,286.480) (270.000,257.680) (514.73312,  -404.07667)  1
A (270.000,257.680) (177.500,216.795) (553.71037,  -509.27917)  1
A (177.500,216.795) ( 85.000,162.220) (628.01623,  -652.46917)  1
A ( 85.000,162.220) (  0.000,100.000) (744.77653,  -828.28417)  1
```

Number of arcs is $N_{arc} = 7$

```
A (500.000,300.000) (407.500,293.155) (500.20991,  -331.25917)  1
A (407.500,293.155) (320.000,274.080) (506.36069,  -370.56000)  1
A (320.000,274.080) (250.000,250.000) (525.32585,  -436.58167)  1
A (250.000,250.000) (190.000,223.120) (556.08512,  -513.63000)  1
A (190.000,223.120) (125.000,187.500) (602.69383,  -607.08750)  1
A (125.000,187.500) ( 65.000,148.620) (669.89912,  -719.13000)  1
A ( 65.000,148.620) (  0.000,100.000) (761.34933,  -850.08750)  1
```

4. Summary

To recap, the problem was how to create a sequence of arcs

- passing through some of the given points and being sufficiently close to the others points,
- arcs must be as long as possible.

We did the following activities:

- examined the problem in the literature,
- developed an algorithm for constructing a sequence of arcs,
- tested our approach with a real data,
- improved the method,
- compared the results.

References

[1] Kazimierz Jakubczyk. Approximation of Smooth Planar Curves by Circular Arc Splines. May 30, 2010 (rev. January 28, 2012)

[2] O. Aichholzer, F. Aurenhammer, T. Hackl, B. Jüttler, M. Oberneder, and Z. Sír. Computational and structural advantages of circular boundary representation.