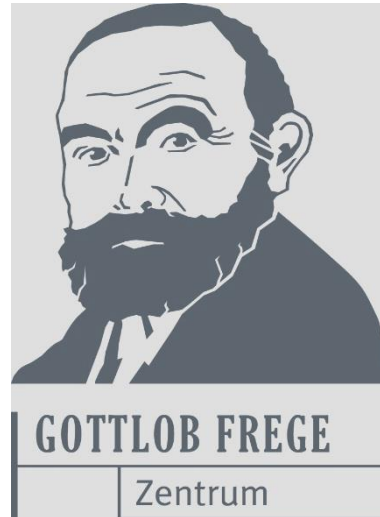


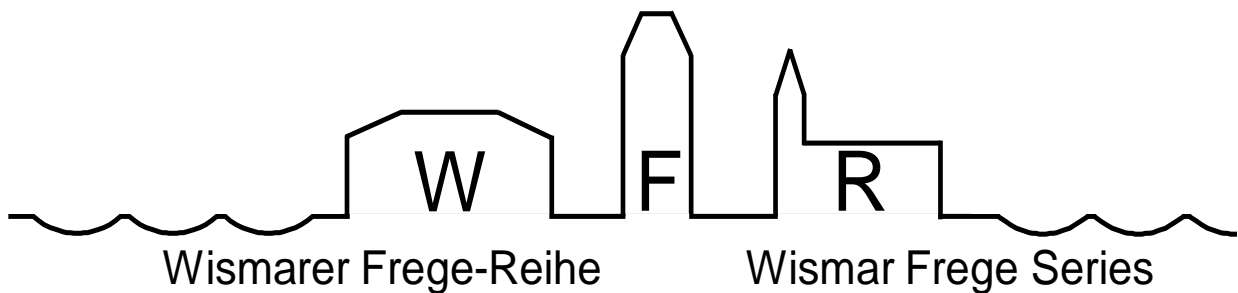
Hochschule Wismar Gottlob Frege Centre



Proceedings
2nd Northern-Light Symposium
on Mathematical Education
in Engineering

Hamburg, September 2019

Heft 03 / 2019



Das **Gottlob-Frege-Zentrum** wurde am 7.11. 2000 an der Hochschule Wismar gegründet. Seine Mitglieder setzen sich für eine wissenschaftlich begründete, praxisorientierte, moderne und international ausgerichtete Ausbildung in den mathematisch-naturwissenschaftlichen Grundlagendisziplinen ein.

Weitere Informationen zum Gottlob-Frege-Zentrum finden Sie auf der Netz-Seite

<http://www.hs-wismar.de/frege>

Die Wismarer Frege-Reihe ist urheberrechtlich geschützt. Eine Vervielfältigung ganz oder in Teilen, ihre Speicherung sowie jede Form der Weiterverbreitung bedürfen der vorherigen Genehmigung durch den Herausgeber.

ISSN 1862-1767

Alle Rechte vorbehalten.

© Hochschule Wismar 2019.
Printed in Germany

WFR Heft 03/2019**Inhalt / Contents**

Vorwort / Preface	2
Einladung mit Programm / Invitation including Program	5
Abendveranstaltung / Evening Event	6

Artikel / ArticlesMathematik und Computer

Thomas Schramm: <i>Maschinelles Lernen und Lineare Algebra – ein didaktischer Ansatz.</i>	8
Thomas Risse: <i>Comparing Mathematical Toolboxes.</i>	14

Blickwinkel Stochastik

Dieter Schott: <i>A nice analytical story about the stochastic Fleishman distributions.</i>	26
Peter Junglas: <i>Is body height really normally distributed?</i>	41

Anhang / Appendix

WFR - Übersicht	51
-----------------	----

Vorwort

Als geborener 49-er machte ich mir nach dem offiziellen Ausscheiden aus dem Berufsleben 2015 so meine Gedanken, welche fruchtbringenden Netzwerke auch in Zukunft noch funktionieren könnten. Als ich im Mai 2001 die IngMath-Workshops in Wismar ins Leben rief (zunächst „Mathematik für Ingenieure“, später „Mathematik in ingenieurwissenschaftlichen Studiengängen“ als Zugeständnis an die inzwischen übliche Geschlechterdifferenzierung bei Berufen), die einen Erfahrungsaustausch zur Mathematiklehre im Ingenieurstudium zum Ziel hatten, suchte ich gleichzeitig nach einem Medium, in dem die Inhalte und Ergebnisse der Workshops „aufgehoben“ (gespeichert) werden. So gibt es seit 2005 die Wismarer Frege-Reihe, in der zu Beginn die Proceedings der Workshops veröffentlicht wurden, später dann aber auch andere Beiträge aus dem Umfeld der Mathematiklehre. Die Vernetzung mit internationalen Institutionen (wie UICEE, WIETE, SEFI Mathematical Workgroup) führte neben deutschsprachigen auch zu englischsprachigen Artikeln. Während die genannten Workshops zunächst an Orten im Norden stattfanden, später in die Mitte und in den Süden wanderten, gab es im Mai 2019 eine Rückkehr an den Ursprung (Außenstelle Rostock-Warnemünde der Hochschule Wismar). Inzwischen geht es wieder südwärts nach Dortmund (Mai 2020). Der Teilnehmerkreis wandelte sich mit den Veranstaltungsorten in gewisser Weise, hatte aber dabei einen festen Stamm. Mir kam nun die Idee, die alteingesessene Nordkonstante mit Peter Junglas (Diepholz), Thomas Risse (Bremen), Thomas Schramm (Hamburg) und mir (Wismar) zu einem zwanglosen Gedankenaustausch einzuladen. Diese Idee fiel bei den anderen genannten Kollegen auf fruchtbaren Boden. Zugleich erklärte sich Thomas Schramm bereit, seine Hamburger Dienststelle (HafenCity Universität) als für alle vier günstig gelegenen Tagungsort zur Verfügung zu stellen. So erfolgte nach einer Terminabstimmung die Einladung zu einem Symposium (Tagung von Wissenschaftlern mit zwanglosen Vorträgen und Diskussionen) am 27.4. 2018, das Thomas Schramm treffend „1st Northern-Light Symposium“ taufte und damit die (zunächst) vier „Nordlichter“ zum Leben erweckte. Besonders reizte Thomas Schramm, dass das ursprünglich altgriechische Wort „Symposion“ ein Gastmahl mit Trinkgelage und Unterhaltung war. Wir fanden das auch reizend.

Peter Junglas machte uns auf das zu unserem Symposium passende Lied „Northern Lights“ (Pulchra est amica mea) für vier Stimmen von Ola Gjeilo (*1978) aufmerksam, das er selbst schon im Kammerchor gesungen hatte und das man sich im Internet auch anhören kann. Ich erinnerte mich, dass ich dieses wunderschöne Lied am 12.7. 2017 abends in der Marienkirche von Ribnitz-Damgarten gehört hatte. Es wurde damals von den Kurpfälzer Madrigalisten gesungen.

Inzwischen hat auch das „2nd Northern-Light Symposium“ am 13.9. 2019 an der HafenCity Universität Hamburg erfolgreich stattgefunden. Die vier dort gehaltenen Vorträge sind in diesen Proceedings in ausgearbeiteter und leicht gekürzter Form (meist in englischer Sprache geschrieben) nachzulesen.

Wir trafen uns diesmal schon vor 14.00 Uhr im Dienstzimmer von Thomas Schramm, um nicht wie beim ersten Mal in Zeitnot zu geraten. Wir begannen wieder mit Kaffee, Kuchen und anregenden Gesprächen. Danach hielten und diskutierten wir unsere Vorträge (siehe Programm S. 5). Sie wurden zwar in Deutsch gehalten, benutzten aber teilweise auch in Englisch formulierte Präsentationen.

Zum Abend fuhren wir in die großräumige Gaststätte „Hobenköök“ (hochdeutsch: Hafenküche) in der Stockmeyerstraße 43 nicht weit von unserem Tagungsort zu einem urgemütlichen Abendessen (siehe S. 6). Es gab ein reichhaltiges Angebot verschiedenster typisch norddeutscher Speisen auf der Grundlage regionaler Angebote. Davon und vom Durstlöscher Bier machten wir mit Freuden und natürlich in Maßen Gebrauch. Angeschlossen war auch eine interessante Verkaufsausstellung mit Nahrungsmitteln und Getränken, die man nicht alle Tage zu Gesicht bekommt.

Während des Abends gab es anregende Diskussionen zu aktuellen Problemen der Mathematikausbildung im Ingenieurstudium an den Hochschulen und zur weiteren Perspektive des Symposiums. Wir waren uns einig, dass die Nordlicht-Symposien eine Fortsetzung verdienen. Angeregt wurde nochmals eine Erweiterung des Teilnehmerkreises um einige Personen, ohne schon eine endgültige Entscheidung zu treffen. Mein Angebot, die Vorträge des Symposiums wieder in der Wismarer Frege-Reihe zu veröffentlichen, wurde gern angenommen. Die Umsetzung liegt mit diesem Heft vor.

Es ist keineswegs vermessen, auch diesmal dem „Vermesser“, Organisator und rührigen Gastgeber Thomas Schramm im Namen aller übrigen Teilnehmer für ein wie immer gut organisiertes und attraktives Symposium nahe am Wasser in der Metropole Hamburg zu danken. Ich hoffe, dass sein Organisations-talent und die HafenCity Universität unserer norddeutschen „Denkfabrik“ noch etliche Jahre zur Verfügung stehen.

Bis zum nächsten Northern-Light Symposium!

Ihr Dieter Schott

Hinweis:

Die Beiträge in den Proceedings haben zum Teil einen anderen Titel als die Vorträge. Auch die Reihenfolge ist hier eine andere. Sie orientiert sich an übergreifenden Themen (vergleiche Inhaltsverzeichnis S. 1 und Einladung S. 5).



Geodäsie & Geoinformatik
Prof. Dr. Thomas Schramm
Überseealle 16
20457 Hamburg
Tel.: 040 42827 5383

EINLADUNG

2nd Northern-Light Symposium on Engineering Education

Liebe Kolleginnen und Kollegen,

Ich möchte Sie herzlich zu unserem zweiten Northern-Light Symposium über mathematische Ausbildung im Ingenieurstudium an Hochschulen am Freitag, 13. September 2019, einladen.

Beginn: 14:00 Uhr, Raum 3.031 HafenCity Universität Hamburg

Programm:

- 14:00-14:30 Prof. Dr. Dieter Schott, Hochschule Wismar: *Wieviel mathematisches Fleisch liefern die Fleishman-Verteilungen aus der Stochastik?*
- 14:45-15:15 Prof. Dr. Thomas Risse, Hochschule Bremen: *Euler Math Toolbox versus MATLAB et al.*
- 15:30-16:00 Prof. Dr. Peter Junglas, Private Hochschule für Wirtschaft und Technik Diepholz: *Are body heights really normally distributed?*
- 16:15-16:45 Prof. Dr. Thomas Schramm HafenCity Universität Hamburg: *Maschinelles Lernen und Lineare Algebra: Ein didaktischer Ansatz.*
- Ab 18:00 Abendprogramm
- Auf Wunsch kann die HCU, speziell das Geo-Lab, VR-Lab, CityScience-Lab oder 3D-Lab, besichtigt werden.

Mit freundlichen Grüßen

Thomas Schramm

Abendveranstaltung

Ein Einblick in das Speiseangebot des Restaurants „Hobenköök“



Quelle: <https://hobenkoeoek.de>

Kommentar: Ist das nicht eine Augenweide? Noch sieht das leckere Essen als Kunstobjekt unversehrt aus. Aber auch Kunst ist nicht für die Ewigkeit gemacht. Mit dem aufkeimenden Appetit wird hier der Kunstliebhaber zum Kunstschänder. Dafür fühlt sich der Gaumen im Paradies.

Ein Ausblick auf das Marktangebot des Restaurants „Hobenköök“



Quelle: <https://hobenkoeoek.de>

Kommentar: Der Markt lädt zum Kauf ein, gerade auch wegen des ungewöhnlichen Angebotes. Bescheidenheit, Trägheit und fehlende Transportkapazitäten zügelten jedoch unseren Zugriff.

Thomas Schramm

Maschinelles Lernen und Lineare Algebra – ein didaktischer Ansatz

Zusammenfassung. Mit dem Projekt „Linear Algebra driven by Data Science“ soll eine Selbstlerneinheit in der Form von „Open Educational Resources (OER)“ geschaffen werden, die mit dem Anwendungsfokus „Maschinelles Lernen“ die Konzepte der Linearen Algebra vermittelt.

Einführung

Lineare Algebra (LA) im MINT-Studium wird klassisch im ersten Studienjahr vermittelt und stellt neben der Differenzialrechnung das Fundament der Höheren Ingenieur-Mathematik dar. Mit der LA werden mathematische Objekte wie Vektoren und Matrizen mit ihren Operationen und Ordnungsbegriffe wie Vektorraum, Dimension, Norm, Metrik mit Konstruktionen wie Basis, Linearkombination, Abbildung, Kern etc. und deren Anwendungen eingeführt, die es ermöglichen komplexe Sachverhalte zu modellieren. Ein in zwei oder drei Dimensionen gewonnenes geometrisches Verständnis kann in der Abstraktion auf höhere Dimensionen und andere Objekte wie Polynome übertragen werden. Die vorherrschende Anwendung ist die Lösung linearer Gleichungssysteme und die analytische Geometrie. Sie bildet die Grundlage für numerische Anwendungen, wie maschinelles Lernen, insbesondere mit künstlichen Neuronalen Netzen.

Das Projekt “Linear Algebra driven by Data Science” trägt der zunehmenden Relevanz des maschinellen Lernens insbesondere der künstlichen Neuronalen Netze als nicht-symbolischen Ansatz Rechnung. Es soll versucht werden, es nicht als nachgeordnete, wenn auch wichtige Applikation zu sehen, sondern als führendes und motivierendes Beispiel, das die Einführung der aufgezählten Objekte, Operationen, Ordnungsbegriffe und Konstruktionen zwingend notwendig macht. Es soll mit einem qualitativen Verständnis für neuronale Netze beginnend, die Konzepte maschinellen Lernens und die benötigten Elemente der Linearen Algebra simultan entwickeln.

Dieses Projekt wird für die Hamburg Open Online University (HOOU, www.hoou.de) von Thomas Schramm, Kay Zobel (HafenCity Universität Hamburg), Ingenuin Gasser, Alexander Lohse (Universität Hamburg), Sören Schwenker (Universität Hamburg, HafenCity Universität Hamburg), Ruedi Seiler und Franz Konieczny (integral learning GmbH, Berlin) entwickelt und implementiert. Die Lerneinheit wird dann über das Portal der HOOU als OER für jedermann zur Verfügung stehen und ist damit ein zweites Modul, das neben der Differenzialrechnung dem Projekt oHMint (online Higher-Mathematics for MINT-Students) zugeordnet und vom sog. OMB+-Konsortium beraten und evaluiert wird (vergl. [1]).

Das Konzept

Die Notwendigkeit, sich mit Linearer Algebra auseinanderzusetzen, wenn maschinelles Lernen professionell betrieben werden soll, wird an vielen Stellen hervorgehoben (vergl. z.B. [2]).

Wie betont, wollen wir aber über ein technisches Verständnis hinaus, eine kritische Grundhaltung vermitteln, die es ermöglicht, die Reichweiten der neuen Technologien des maschinellen Lernens (ML) jenseits der Hochglanzbroschüren einzuschätzen und mögliche Risiken einzuordnen. Fragen, die in diesem Zusammenhang erläutert werden sollen sind z.B.

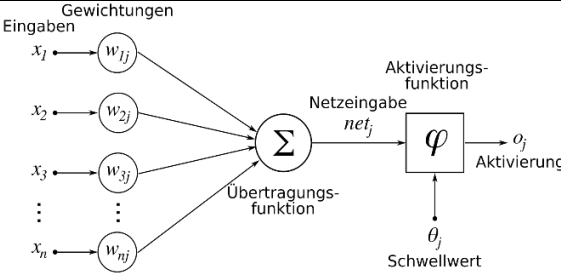
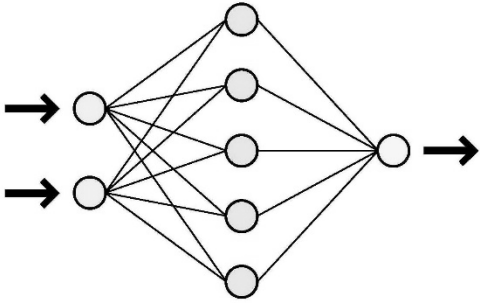
- Wer ist für Entscheidungen verantwortlich, die auf der Basis von ML getroffen werden?
- Lassen sich ML-Entscheidungen immer verstehen?
- Wo und wie ist das gelernte Wissen repräsentiert?

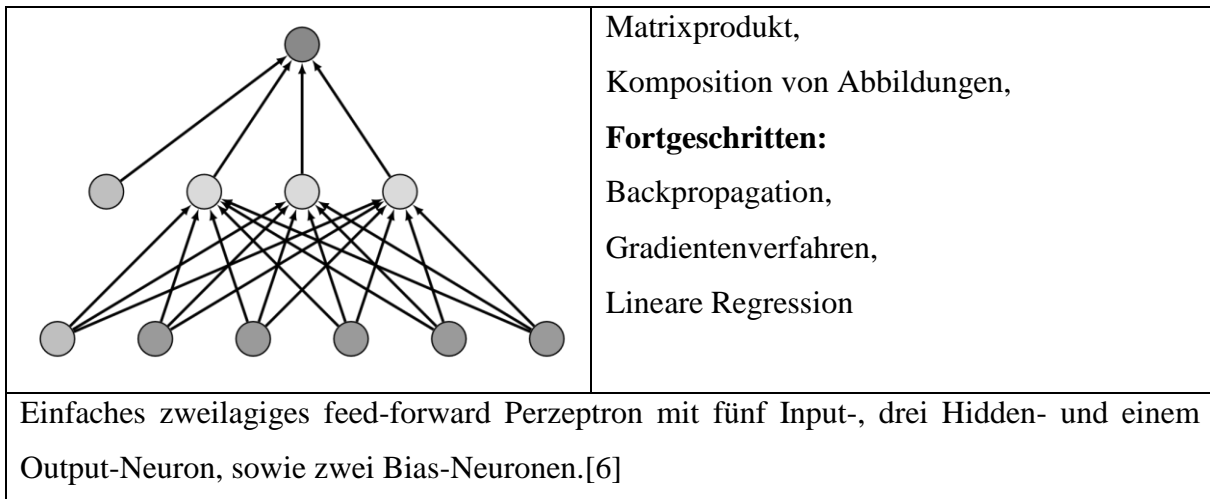
Hierbei müssen die verwendeten allgemeinen Begriffe wie *Data Mining*, *künstliche Intelligenz*, *maschinelles Lernen*, *überwachtes* versus *unüberwachtes Lernen*, *klassifizieren* (Support Vector Machines) etc. erst einmal geordnet und zumindest qualitativ eingeführt werden.

Im Weiteren beschäftigen wir uns dann mit der wesentlichen Methode des nicht-symbolischen maschinellen Lernens durch künstliche Neuronale Netze (kNN). Mit einem qualitativen, modellhaften Verständnis für die Funktionsweise neuronaler Netze beginnend, werden deren Begrifflichkeiten, deren algorithmische Umsetzung und die genutzten mathematischen Methoden parallel entwickelt.

Im Detail

Hier soll keine Einführung in künstliche neuronale Netze gegeben, sondern nur die begrifflichen Zusammenhänge zur Linearen Algebra exemplarisch und tabellarisch dargestellt werden. Für eine Einführung siehe z.B. [3]

Begriffe: künstliches Neuronales Netz	Begriffe: Lineare Algebra
 <p>Gewichtungen Eingaben $x_1 \rightarrow w_{1j}$ $x_2 \rightarrow w_{2j}$ $x_3 \rightarrow w_{3j}$ \vdots $x_n \rightarrow w_{nj}$</p> <p>Übertragungsfunktion Σ Netzeingabe net_j Aktivierungsfunktion φ Aktivierung o_j Schwellwert θ_j</p>	<p>Vektor, Vektorraum, Linearkombination, Skalarprodukt</p>
Schema für ein künstliches Neuron. [4]	
	<p>Matrix, Matrix-Vektorprodukt, Matrix-Abbildung, Rang, Diagonalisierung, Bild, Kern, Dimension, Lineare (Un)-Abhängigkeit, Lineare Gleichungssysteme</p>
Einfaches Perzeptron. Vereinfachte Darstellung eines künstlichen neuronalen Netzes. [5]	



Anwendung und informatisches Grundwissen

Die zu entwickelnde OER soll passive aber auch interaktive Elemente enthalten. Hierzu zählen Videos und Animationen, die Beispiele erläutern und sich ggf. steuern lassen. Ein gestuftes formatives Assessment ist obligatorisch, um den Lernfortschritt für die Lernenden transparent zu gestalten und entsprechend anonymisiert den Entwicklern Aufschluss über den Erfolg gibt. Aus dem oHMint-Projekt liegen bereits bereits Erfahrungen mit dem Einsatz solcher audiovisuellen Stilelemente vor, die auch durch kleine Wettbewerbe (Gamification) bereichert werden können.

Neben der Einführung der Konzepte soll ein Beispiel mit wachsender Komplexität parallel entwickelt werden. Hier kann es sich z.B. darum handeln, in einer 4x4-Matrix ein Kreuz von einem Kreis zu unterscheiden, bis hin zum Standard-Beispiel der Zahlenerkennung einer handschriftlichen Eintragung, für das umfangreiche Trainingsdaten verfügbar sind.

Innerhalb der Lernumgebung wird es möglich sein, kleine Code-Schnipsel zu implementieren und zu testen, um diese später zu einem komplexen Beispiel zusammen zu fügen. Der Fokus wird aber nicht darauf liegen, einfach nur vorhandene Bibliotheken zu verbinden, sondern ganz elementar die benötigten Datenstrukturen (Vektoren, Matrizen) und Operationen (Matrizenaddition und –multiplikation) selbst zu definieren, um diese dann ggf. erst später durch entsprechende Bibliothekselemente zu ersetzen. Die Realisierung kann in Python

oder Octave bzw. Matlab[®] erfolgen, die in die Lernumgebung eingeflochten werden.

Der wünschenswerte Nebeneffekt ist natürlich die Entwicklung zumindest elementarer Kompetenzen im Bereich der Programmierung.

Fazit

Skizziert wurde die Entwicklung einer Selbstlerneinheit in der Form einer OER, die das aktuelle „heiße“ Thema der künstlichen Intelligenz bzw. des maschinellen Lernens aufgreift und motivieren soll, sich mit den notwendigen Werkzeugen aus der Linearen Algebra auseinander zu setzen. Gleichzeitig werden grundlegende Konzepte aus der Informatik erarbeitet, die die praktische Implementierung der gewonnenen Erkenntnisse ermöglicht.

Die fertige Lerneinheit wird auf der HOOU-Plattform für die interessierte Zivilbevölkerung angeboten werden, wird aber gleichzeitig bereits während der Entwicklung im universitären Umfeld als Blended-Learning oder im Flipped-Classroom-Kontext genutzt bzw. evaluiert.

Der sonst so schwierige Ansatz Mathematik projektorientiert und erfahrbar zu vermitteln, weil die Projekte sonst entweder trivial oder zu komplex sind, kann hier gelingen.

Literatur

- [1] Barbas H., Gasser I., Konieczny F., Lohse A., Seiler R.: oHMint: Online Higher Mathematics for MINT Students - An Online Mathematics Course and Learning Platform, Proceedings of the 46th SEFI Annual Conference, 2018, Copenhagen, Denmark.
- [2] Donges N.: Basic Linear Algebra for Deep Learning, towards data science, 2018, towardsdatascience.com/linear-algebra-for-deep-learning-f21d7e7d7f23
- [3] Frochte J.: Maschinelles Lernen, Carl Hanser Fachbuchverlag, 2019
- [4] Kein Autor, unverändert, CC BY-SA 3.0
commons.wikimedia.org/w/index.php?curid=674079
- [5] Dake, Mysid, unverändert, CC BY 1.0,
commons.wikimedia.org/wiki/File:Neural_network.svg
- [6] Thoma M., Eigenes Werk, unverändert, CC BY 3.0,
commons.wikimedia.org/w/index.php?curid=27872538

Autor

Prof. Dr. Thomas Schramm

HafenCity Universität Hamburg

Überseeallee 16

D-20457 Hamburg

E-Mail: thomas.schramm@hcu-hamburg.de

Thomas Risse

Comparing Mathematical Toolboxes

Abstract. Comparing mathematical toolboxes on one hand gives some insight in potential and limits of such tools for engineers and on the other hand leads to ideas on the ideal Math Toolbox with which existing computer support is competing. To this end we mainly will compare MATLAB and its Symbolic Math Toolbox, SMT, based on MuPad with Euler Math Toolbox, EMT, and its symbolic computations based on Maxima in order to get insight into general characteristics of suchlike computer support.

Computer supported Engineering

There is a manifold of systems supporting engineering math. Some are proprietary like the three big 'M', Maple [3], Mathematica [4] and MATLAB [5], some are open source like SAGE [10], Octave [8] and SciLab [11], open source alternatives to MATLAB, or Euler Math Toolbox, EMT [1]. In general they combine scientific, i.e. numerical and symbolical computations. Typically the user communicates with the system by either some proprietary (scripting) language as in the case of e.g. the three big 'M' or by a general programming language like e.g. Python as in the case of SAGE. Due to the impressive potential of such systems comprehensive documentation, adaptive help system and meaningful error reporting are essential. In addition, most systems are to be expanded by packages for special applications.

Hence, assessing such computer support the systems are to meet a set of criteria addressing so divers properties as

- numerical and symbolical potential/power made available to engineering,
- user interface with ease of use, conformity to expectation, with integrated help system and self-explanatory error reporting,
- performance/robustness/scalability/extendability,
- Continual Improvement Process by a company or a community,

- preferably open source so that the software is easily deployed also in less developed countries.

Here, we will compare MATLAB [5] with Euler Math Toolbox, EMT [1]. The two systems are rather similar insofar as they offer the same data types like double vectors, double matrices, sparse matrices, strings with more or less the same initialisers like `zeros`, `ones`, `eye`, `diag`, `linspace`, etc. and the same set of operators like `size`, `det`, `rank` and the backslash operator to numerically solve systems of linear equations.

User Interface

The user communicates with MATLAB and EMT by very similar commands. Both systems can be programmed using a mostly procedural language with the usual control structures `for`, `while`, `if`: only MATLAB features a `switch` statement – only EMT offers `repeat` and a simple and fast `loop` statements.

In both systems the main data type is (vectors and matrices of) IEEE 754 doubles and in both systems the colon operator produces vectors of equally spaced numbers or indices. In EMT, closed intervals with double boundaries exhibit a data type which MATLAB natively does not support.

MATLAB and EMT implement the same concepts of variables, assignments, scope of variables, and functions. There are minor differences as for example in EMT the body of functions is to be terminated by `endfunction` as e.g. in SciLab.

Whereas in MATLAB vectorisation is explicitly done by the dot operator it is default in EMT. In addition, EMT sports `map` to easily vectorise functions.

MATLAB `help` and `doc` commands provide convenient help to operators, control structures and commands. EMT's help information is not as comprehensive; however, every keyword can be clicked to get to the corresponding help information. Both systems provide helpful (user) error messages where MATLAB even represents the call sequence leading to the error.

Numerical Computations

Both SMT and EMT offer a wide range of built-in functionality for scientific computations like

- solving systems of linear equation with partial pivoting (MATLAB and EMT: backslash operator, LU – EMT: `lusolve`)
- solving linear least squares problems (MATLAB: backslash operator, `polyfit`, `lscov` via Cholesky and QR – EMT: `fitnormal`, `fit`, `svdsolve`)
- numerical differentiation (MATLAB: `diff` – EMT: `diff`, `diffc`)
- numerical integration, i.e. quadrature (MATLAB: `integral`, `integral2`, `integral3`, `trapz` – EMT: `simpson`, `gauss`, `romberg`, `integrate` adaptive Gauss, `gausstwo`)
- zero finding (MATLAB: `roots`, `fzero` combination of bisection, secant, and inverse quadratic interpolation – EMT: `polysolve`, `polyroot`, `bisect`, `solve` secant, `newton`, `fzeros`, `broyden`)
- interpolation/approximation (MATLAB: – EMT: `divdif` divided differences, `spline`, `hermiteinterp`, `remez` Remez algorithm)
- optimization, i.e. approximating minima (MATLAB `fminbnd` golden section and parabolic interpolation, `fminsearch` Nelder-Mead, in optimization toolbox, `lsqnonlin` – EMT: `fextrema`, `fmax/fmin` golden cut, `neldermin`, `brentmin`, `powellmin`)
- solving ordinary differential equations (MATLAB: `ode23`, `ode45` Runge-Kutta – EMT: `heun`, `runge`, `adaptiverunge`)
- solving partial differential equations (MATLAB: with PSE toolbox – EMT: (ordinary) differential equations [2] are solved using the Laplace transform.)
- performing transformations (MATLAB: `fft`, `fft2` – EMT: `fft`, `fht` Hartley)

Although both MATLAB and EMT offer algorithms for the most pertinent numerical problems in general, MATLAB seems to be more user friendly as for example MATLABs `fzero` uses a beneficial combination of bisection, secant, and inverse quadratic interpolation methods whereas EMT provides each algorithm and leaves the selection at the discretion of the user.

Let us compare the performance of MATLAB and EMT by the time it takes each tool to generate random coefficient matrices and random vectors of the right hand side and then to solve the corresponding linear equation. It also returns the norm of the residuum. This benchmark also shows the similarity of the code in MATLAB and EMT.

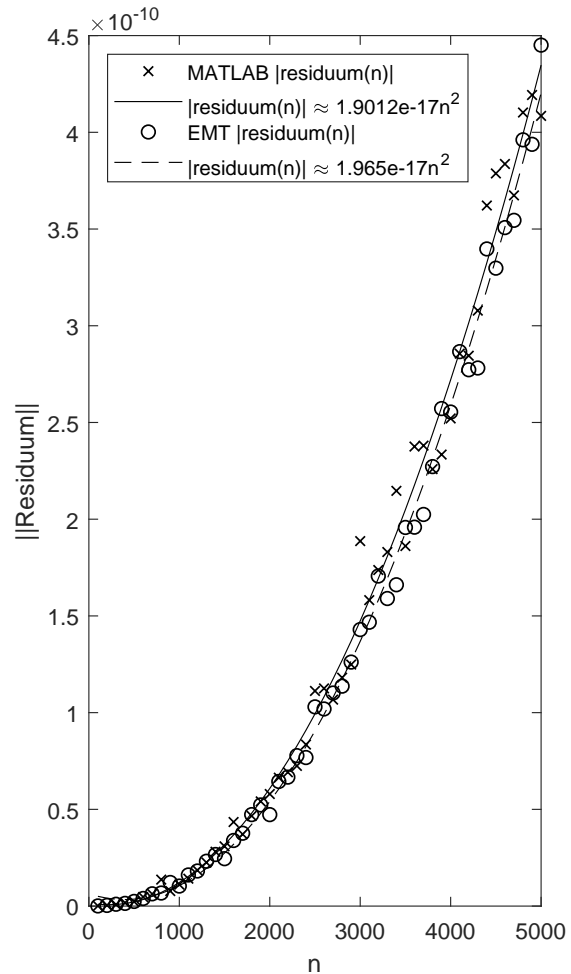
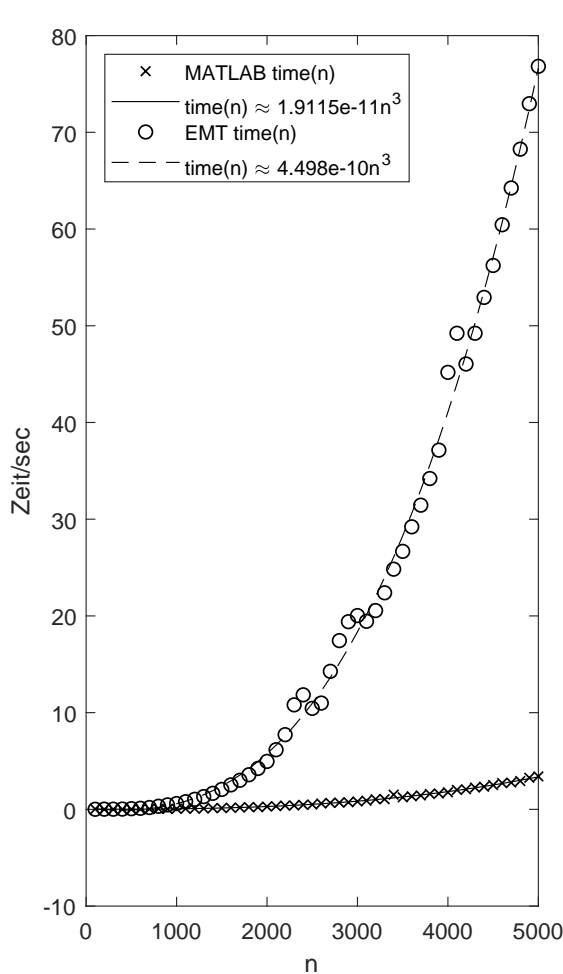
In order to compare the results we compute least squares polynomials to approximate the execution times of the benchmark and the norm of the residua depending on the number n of variables. We use a cubic polynomial for the execution times because the number of floating point operations

needed to execute the Gauss algorithm is proportional to n^3 . A quadratic polynomial seems to be a good fit for the norm of the residua. Using for example a cubic polynomial results in a negative leading coefficient corroborating this choice of model function.

On one hand, it turns out that EMT is twentyfold slower than MATLAB. Considering that EMT uses one processor kernel only whereas MATLAB uses at least two of four, EMT is at least in this benchmark about ten times slower than MATLAB. On the other hand, the resulting residuum norms have the same order of magnitude as can be seen also comparing the leading coefficients of the two quadratic regression polynomials.

```
tic; A = rand(n); x = rand(n,1);
b = A*x; xx = A\b;
residuumnorm = norm(A*xx-b); t = toc;
% return [t,residuumnorm]
```

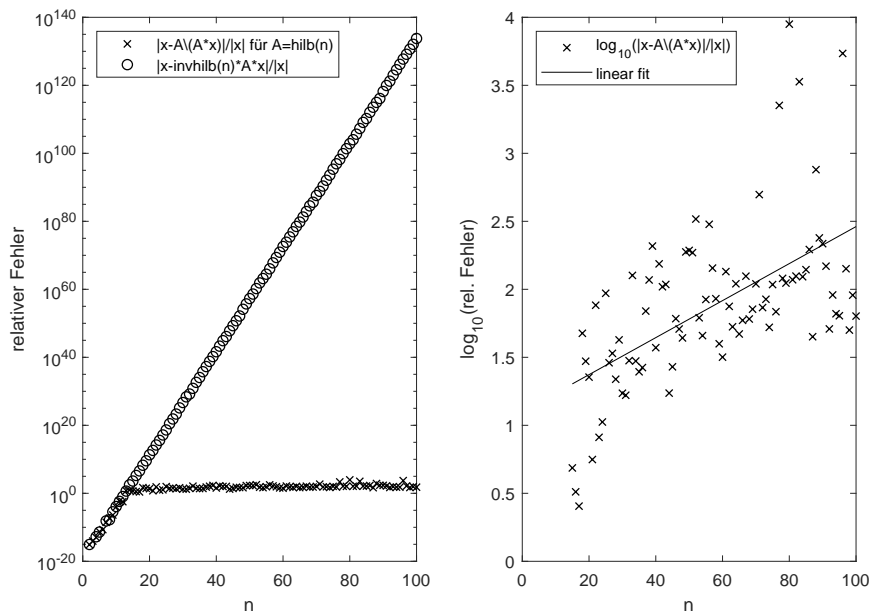
```
tic; A = random(n,n); x = random(n,1);
b = A.x; xx = A\b;
residuumnorm = norm(A.xx-b); t = toc;
return [t,residuumnorm];
```



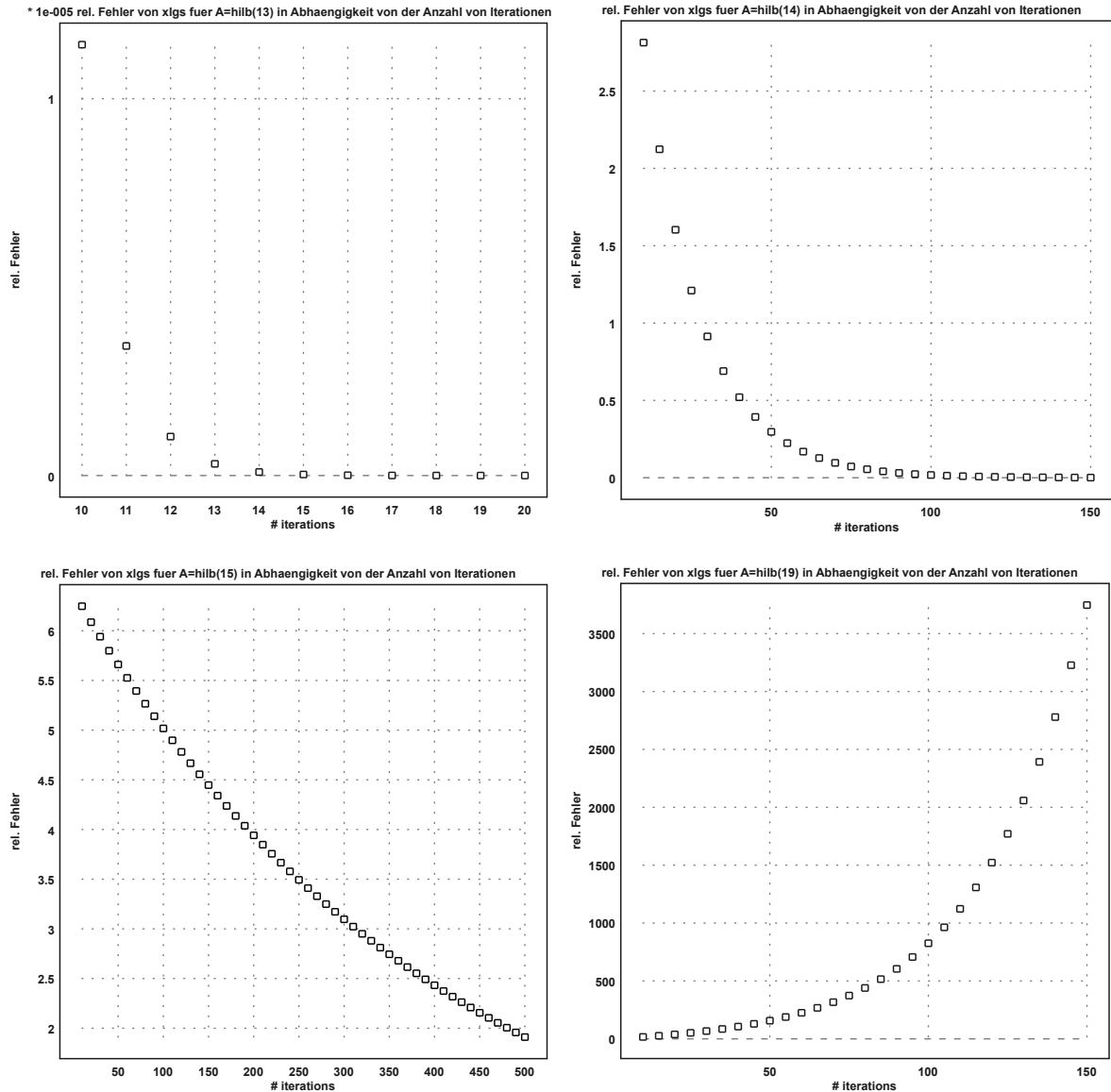
It took EMT rather long to compute the more appropriate relative residual $\frac{\|\mathbf{x}\|}{\|\mathbf{A}\| \|\mathbf{x}\|}$ because in case of the Euclidean norm the evaluation of the matrix norm involves a singular value decomposition or the computation of the eigenvalues of $\mathbf{A}'\mathbf{A}$.

To compare the accuracy of solutions of linear equations we use the notorious example of the (unscaled) Hilbert matrix $\mathbf{H}_n = (h_{i,j})$ with $h_{i,j} = \frac{1}{i+j-1}$ for $i, j = 1, 2, \dots, n$ in $\mathbf{H}_n \mathbf{x} = \mathbf{b} := \mathbf{H}_n \mathbf{1}$.

MATLAB provides a function `invhilb` to invert `hilb(n)`. Comparison shows that the inverse is only helpful for rather small $n < 12$:



In addition to the backslash operator, EMT provides the function `xlgs` to improve solutions of linear equations by residuum iteration and an exact scalar product using a long accumulator. `xlgs` works very well on scaled \mathbf{H}_n for small $n \leq 12$ but mostly fails for bigger n as the examples $n = 13$ (fast convergence), 14 (fast improvement), 15 (slow improvement), 19 (deterioration) below demonstrate. The four cases show that there is no way to a priori know how many iterations will achieve a given accuracy.



These examples leave the question why not solve aggravating problems like $\mathbf{H}_n \mathbf{x} = \mathbf{b}$ symbolically in the first place. The next section illustrates potential and pitfalls of this approach.

Symbolical Computations

A computer algebra system, CAS is designed to solve problems symbolically. For example it simplifies algebraic expressions, evaluates elementary mathematical functions on symbolical arguments, solves equations symbolically, differentiates, integrates, transforms etc. Symbolical results are exact results although sometimes not really helpful (e.g. solve ...) or utterly wrong as there is no correct/flawless/bugless software.

One has to distinguish four cases. Either a CAS solves a problem (ok) or

gives only a partial solution (with or without indicating that for example there are/might be other solutions), or indicates failure typically by returning the problem as stated or worst of all produces wrong answers. The two systems under consideration present many examples of this behaviour.

To compare CASs one needs criteria, in this case a set of mathematical problems. Here we use Westers set of problems [12] from 1999 which cover many areas of mathematics. Of course, the relevance/pertinence of any of these problems can be questioned. As with any benchmark there is no fair set of criteria! However, Westers set of problems allows not only to compare CASs but also to check whether versions of some CAS improve.

In MATLAB the Symbolical Math Toolbox, SMT is based on the computer algebra system MuPad, the Multi Processing Algebra Data Tool developed at the University of Paderborn. Users either define symbolical variables by `sym` or `syms` and natively operate on these for example by overloaded versions of the elementary mathematical functions; or they call MuPad directly by passing strings of commands to it. SMT converts the results, returned as strings, to proper symbolical values. If SMT or MuPad fail to solve a problem MATLAB might return a numerical solution. Similarly, symbolical computations in Euler are based on the computer algebra system Maxima, an offspring of Macsyma developed at MIT. However the interaction between EMT and Maxima is ...

In [9] SMT and EMT are compared by some problems from [12] ordered in well over two dozen areas of mathematics. Here we order some findings according to the type of answers given by the CASs.

correct 😊 There are problems SMT can solve and EMT cannot and vice versa as is to be expected considering the complexity of CASs and the diversity of mathematical problems.

- So for example SMT cannot do C4, i.e. hex to integer conversion whereas EMT can,
- SMT can compute a standard deviation symbolically, E5, whereas EMT cannot.
- MuPad, not SMT can solve linear equations in residue class rings whereas EMT cannot.
- SMT does not simplify H1, i.e. $2 \cdot 2^n$ and H2, i.e. $4 \cdot 2^n$ to 2^{n+1} and 2^{n+2} respectively whereas EMT does.
- Factoring H15, i.e. $x^3 + x^2 - 7$ and expanding the result should obviously return H15, but SMT fails (even though there are symbolical formulae for the three zeros of H15) whereas EMT succeeds

when using `solve` and `prod`.

- SMT knows special values J7, J8 of Bessel's J function whereas EMT does not.

Sometimes a CAS can produce a correct answer only with some *user* assistance, e.g.

- SMT cannot solve T3, i.e. $\lim_{x \rightarrow \infty} \frac{\log x}{\log x + \sin x} = 1$ but it solves $\text{limit}(1/(1+\sin(x)/\log(x)), x, \text{inf})$.

- Similarly EMT fails at W5, i.e. $\int_0^2 \sqrt{x + \frac{1}{x} - 2} dx = \frac{8-\sqrt{8}}{3}$, but succeeds on $(\int_0^1 + \int_1^2) \sqrt{x + \frac{1}{x} - 2} dx$ when the *user* splits the integral at the singularity of the integrand.

partially correct ☹️ • SMT can solve $3x^3 - 18x^2 + 33x - 19 = 0$ in \mathbb{R} only numerically. It might miss complex solutions. In contrast, EMT computes three solutions but is unable to identify at least one real solution.

- SMT and EMT fail at periodic solutions.

[12]	SMT R2019a	MXM 5.43
M13: $\text{solve}(\sin x = \cos x) \leftrightarrow x \in \frac{\pi}{4} + \pi\mathbb{Z}$	$\leftrightarrow \text{pi}/4$ ☹️	trigsimp, solve ☹️
M14: $\text{solve}(\tan x = 1) \leftrightarrow x \in \frac{\pi}{4} + \pi\mathbb{Z}$	$\leftrightarrow \text{pi}/4$ ☹️	$\leftrightarrow \text{pi}/4$ ☹️
M15: $\text{solve}(\sin x = \frac{1}{2}) \leftrightarrow x \in \{\frac{\pi}{6}, \frac{5\pi}{6}\} + 2\pi\mathbb{Z}$	$\leftrightarrow [\text{pi}/6, 5*\text{pi}/6]$ ☹️☹️	$\leftrightarrow \text{pi}/6$ ☹️

- Both SMT and EMT miss the other solution of M31, i.e. solve $1 - |x| = \max\{-x - 2, x - 2\}$ which gives $x = \pm\frac{3}{2}$.

- SMT cannot show W14, i.e. $\int_{-\infty}^{\infty} \frac{\sin t}{t} e^{2it} dt = 0$ but it does show that both real and imaginary part vanish.

failed attempt 😞 Here we present some of the examples [9] of [12] in which both CASs fail, namely

- D5: to compute cubic splines symbolically,
- F4: to simplify $2^n n! \prod_{k=1}^n (2k - 1)$ to $(2n)!$,
- I2: to simplify $\sqrt{\frac{1+\cos 6}{2}}$ to $-\cos 3$
- K25: to simplify $w + \sqrt{w+1}\sqrt{w-1}$ to z or $\frac{1}{z}$ for $w := \frac{z+1/z}{2}$,
- L5, L6: to reduce $\log(\tan(\frac{x}{2} + \frac{\pi}{4})) - \text{arsinh}(\tan x)$ as well as its derivative to 0,
- N17: to solve $(x + y > 0, x - y < 0)$ to give $|x| < y$,
- P39: to compute the singular value decomposition of $\begin{pmatrix} 1 & 2 & 3 \\ 1 & 2 & 3 \end{pmatrix}^\top$ symbolically,
- to simplify certain finite sums (R3 and R10) and certain infinite

sums (R21 and R24) involving binomial coefficients,

- to simplify certain infinite products (S7 to S10),
- Z22: to solve the system $\dot{x} = x(1 + \frac{\cos t}{2 + \sin t}), \dot{y} = x - y$ of ordinary differential equations,
- to solve partial differential equations (e.g. a1 and a2),
- to evaluate Legendre polynomials (c7 to c9).

wrong answer ⚡ In several cases SMT or EMT produce wrong results. This is most disturbing ??? because any such error undermines the all over credibility of a CAS. To what extend can one trust a CAS which from time to time and without warning produces wrong answers.

- EMT cannot combine intervals and symbolic variables in D12, i.e. expanding $([-4, 2]x + [1, 3])^2$.
- SMT fails at this problem as there is no built in interval arithmetic anyway.
- EMT fails to show that for $x < 0$ K10, i.e. $\sqrt{\frac{1}{x}} + \frac{1}{\sqrt{x}}$ vanish as SMT returns, instead EMT correctly computes $\frac{2}{\sqrt{x}}$.
- SMT cannot solve the symbolic (in)equality M31, i.e. $(1 - |x| = \max\{-x - 2, x - 2\})$ which should return $\pm \frac{3}{2}$.
- From the assumption N8, i.e. $x \geq y \geq z \geq x$ SMT does not deduce $x = y = z$. However, EMT does deduce this relation.
- EMT fails at computing the norm of O1, i.e. $(1 + i, -2, 3i) \in \mathbb{C}^3$, SMT correctly returns $\sqrt{15}$.
- Both CASs compute the rank of the two matrices P16, i.e. $\begin{pmatrix} 2\sqrt{2} & 8 \\ 6\sqrt{6} & 24\sqrt{3} \end{pmatrix}$ and P17, i.e. $\begin{pmatrix} \sin(2\theta) & \cos(2\theta) \\ 2\sin^2\theta \cos\theta & \cos(2\theta) \sin\theta \end{pmatrix}$ wrongly: both ranks are 1 and not 2. Here simplification obviously does not succeed in the Gauss algorithm.
- SMT fails to compute W15, i.e. $\int_0^1 \ln \Gamma(x) \cos(6\pi x) dx = \frac{1}{12}$ whereas EMT erroneously ??? reckons the integral to be divergent.

Please allow for a caveat: the CASs generated the above results when used as is. If one is willing to experiment with more simplification steps, with several calls to simplify or with reformulations of the problem at hand then results sometimes would differ, e.g. I3, I4 [9]

There are more intricate differences of SMT and EMT.

simplification EMT simplifies C2, i.e. `factor(50!)` to `50!` because it

considers Maxima's result $2^{47} \cdot 3^{22} \cdot 5^{12} \cdot 7^8 \cdot 11^4 \cdot 13^3 \cdot 17^2 \cdot 19^2 \cdot 23^2 \cdot 29 \cdot 31 \cdot 37 \cdot 41 \cdot 43 \cdot 47$ less simple. The surprised EMT user has to ask for Maxima's result explicitly.

EMT/Maxima offers a wide range of functions for simplification like `ratsimp`, `partfrac`, `trigsimp` or `radcan` and options like `trigexpand`, `trigreduce`, `logcontract`, `distrib` or `rootscontract`. Of course, in EMT users can tailor simplification to their needs. But in general it is deemed best and convenient to let the CAS do the simplification on its own as is the case in SMT.

assumptions It would be very convenient if a CAS would respect more complicated assumptions, but both CASs fail already at a simple assumption H19: `assume(a2 = 2)` so that $\frac{1}{a-1}$ should be simplified to $a + 1$.

Some assumptions seem not to be representable, e.g. K8, i.e. $\sqrt{\frac{1}{z}} - \frac{1}{\sqrt{z}} = 0$ for $z \in \mathbb{C} \setminus \mathbb{R}^-$ because algebraic or logic expressions like `assume((real(z)<0)&&(imag(z)=0)==0)` are not allowed in assumptions in SMT and EMT.

missing data types O3, i.e. the Lagrange identity $\mathbf{a} \times \mathbf{b} \cdot (\mathbf{c} \times \mathbf{d}) = (\mathbf{a} \cdot \mathbf{c})(\mathbf{b} \cdot \mathbf{d}) - (\mathbf{b} \cdot \mathbf{c})(\mathbf{a} \cdot \mathbf{d})$ and O5, i.e. divergence of a vector product $\operatorname{div}(\mathbf{f} \times \mathbf{g}) = \mathbf{g} \cdot \operatorname{rot} \mathbf{f} - \mathbf{f} \cdot \operatorname{rot} \mathbf{g}$ or $\nabla \cdot (\mathbf{f} \times \mathbf{g}) = \mathbf{g} \cdot (\nabla \times \mathbf{f}) - \mathbf{f} \cdot (\nabla \times \mathbf{g})$ make it necessary to be able to specify a symbolic variable to be a vector in \mathbb{R}^2 (`syms v 'plane'`) or in \mathbb{R}^3 (`syms w 'space'`) if one is not willing to work with $\mathbf{v} = (v_x, v_y)$ and $\mathbf{w} = (w_x, w_y, w_z)$.

Here SMT and EMT are beaten by CASs like SAGE [10] that can handle data types like \mathbb{Q} , $\mathbb{GF}(p^m)$, Gaussian and Eisenstein integers, etc. and vectors, matrices, polynomials over finite and infinite fields.

In total, above all erroneous answers undermine the applicability of any CAS. It is the foremost obligation of developers and maintainers to eradicate said errors.

Conclusion

In a way, our observations lead to something like a wish list for mathematical toolboxes for engineering.

- user/problem orientation
- syntax aligned with mathematical convention

- conformity to expectation, sensible defaults, no hidden parameters
- homogeneity of commando syntax (cp. e.g. `rand`, `ones`, `zeros`)
- arithmetic in $\mathbb{Q}, \mathbb{R}, \mathbb{C}, \mathbb{GF}(p^m)$, towers of data types, strong typing
- orthogonality as in Instruction Set Architectures, ISAs, e.g. `gcd(r, s)` for $r, s \in \mathbb{N}$ as well as for $r, s \in \mathbb{GF}(p^m)[x]$
 \Rightarrow object orientation / domain detection / coercion, cp. e.g. SAGE
- integration of numerical and symbolical computing and interval arithmetic
- assumptions on expressions, cp. e.g. SAGE
- robustness / performance / scalability
- syntax highlighting / command completion / function tool tips
- integrated help system with pertinent examples / error reporting
- open source / Continual Improvement Process / community

Bibliography

- [1] *Euler Math Toolbox, EMT*. <http://euler-math-toolbox.de>
 = **Grothmann, R.:** Euler Math Toolbox <http://euler.rene-grothmann.de>
- [2] **Grothmann, R.:** *Solving ODE with Laplace*. <http://euler.rene-grothmann.de/Programs/Examples/SolvingODEwithLaplace.html>
- [3] *Maple, the essential math tool*. <https://de.maplesoft.com>
- [4] *Mathematica, the system for technical computation*. <https://www.wolfram.com>
- [5] *MATLAB, a programming platform designed specifically for engineers and scientists*. <https://mathworks.com>
- [6] *Maxima, a Computer Algebra System*. <http://maxima.sourceforge.net>
- [7] *MuPAD, the Multi Processing Algebra Data Tool*, see e.g. The Mathworks on SMT and MuPad <https://de.mathworks.com/discovery/mupad.html>
- [8] *Octave*. <https://www.gnu.org/software/octave/>
- [9] **Risse, Th.:** *Euler vs MATLAB et al.* 2nd Northern-Lights Symposium on Engineering Education, HCU, September 13th, 2019
www.weblearn.hs-bremen.de/risse/papers/NLSoEE_2_HCU_2019/Euler_vs_MATLAB.pdf

- [10] SAGE = System of Algebraic and Geometric Experimentation
<http://www.sagemath.org>
- [11] SciLab – a free and open source software for engineers & scientists
<https://www.scilab.org>
- [12] **Wester, M.J.:** *A Critique of the Mathematical Abilities of CASs.* in **Wester, M.J. (ed):** *Computer Algebra Systems – A Practical Guide.* Wiley 1999
https://math.unm.edu/~wester/cas_review.html

Author

Prof. Dr. rer. nat. Thomas Risse
Fakultät E-Technik & Informatik
Hochschule Bremen – City University of Applied Sciences
Flughafenallee 10, D-28199 Bremen
E-Mail: risse@hs-bremen.de

Dieter Schott

A nice analytical story about the stochastic Fleishman distributions

Summary: The Fleishman system of distributions is often used in simulation experiments to generate non-normal distributions with given skewness γ_1 and excess γ_2 . A Fleishman distribution is generated from a standard normal random variable x (mean $\mu = 0$ and variance $\sigma^2 = 1$) by

$$y = a + bx + cx^2 + dx^3$$

with real parameters a, b, c and d . For appropriate pairs (γ_1, γ_2) members of the system exist. The coefficients can be calculated from a nonlinear system of algebraic equations. Solvability conditions and properties of solutions are discussed.

1. Introduction

For a thorough study concerning the effectivity and robustness of statistical selection methods of Bechhofer and Gupta type large simulation experiments are needed using also non-normal distributions [5]. It turned out that so-called *Fleishman distributions* with four parameters introduced in [1] are a suitable class of such distributions. In [1] and some other sources (see e.g. the textbooks [3], [4]) equations for the parameters of Fleishman distributions are given and partly tables of calculated parameters for prescribed first four moments are listed. Since these equations contain differences in signs and the corresponding tables are not extensive enough as well as partly suggest wrong conclusions we decided to check the equations and tables to guarantee a correct simulation study. Further, we need some software to substitute and extend the existing tables for the coefficients of the Fleishman distributions. Such software is available in the meantime using the statistical package R. Here we use MATLAB to produce solutions and to investigate the Fleishman equations. The presentation [6] is based on this paper.

2. The Fleishman Distributions

We consider random variables

$$y = y(a, b, c, d) = a + bx + cx^2 + dx^3 \quad (1)$$

with real coefficients a, b, c and d , where x is standardized normally distributed ($x: N(0,1)$). Hence, there are four unknown parameters. The variables y create so-called *Fleishman distributions*. This family contains apart from derivatives of $N(0,1)$ for $c = d = 0$ mainly non-normal distributions. As mentioned in [1] the

chi-quadrat distributions with one degree of freedom are a subclass which arises for $d = 0$.

If we additionally prescribe the mean (expectation) $\mu = E(\mathbf{y}) = 0$ and the variance $\sigma^2 = V(\mathbf{y}) = 1$ we obtain standardized Fleishman distributions.

We can expect to get single Fleishman distributions if we add to the conditions $\mu = 0$ and $\sigma^2 = 1$ two further conditions. Namely, then we have four equations for the four coefficients in (1). It is convenient to choose the skewness $\gamma_1 = E(\mathbf{y}^3)$ and the excess $\gamma_2 = E(\mathbf{y}^4) - 3$ which are both 0 for $N(0,1)$.

The general solvability of this algebraic system of equations is considered. Although in the references [2] and [3] only solutions with positive coefficients a, b, c, d occur, the properties of distributions connected with the other real solutions are investigated, too.

We investigate this family $FD(0,1,\gamma_1,\gamma_2)$ of Fleishman distributions with two parameters. It is worth to mention that all distributions with existing first four moments have to fulfil *Pearson's inequality*

$$\gamma_2 \geq \gamma_1^2 - 2. \quad (2)$$

Consequently the admissible pairs (γ_1, γ_2) are at least restricted to an area inside the parabola $\gamma_2 = \gamma_1^2 - 2$.

By simulation you can get some impression about possible shapes of corresponding Fleishman probabilistic density functions (p.d.f.). Here we show two special cases in Figure 1. The second one is not unimodal.

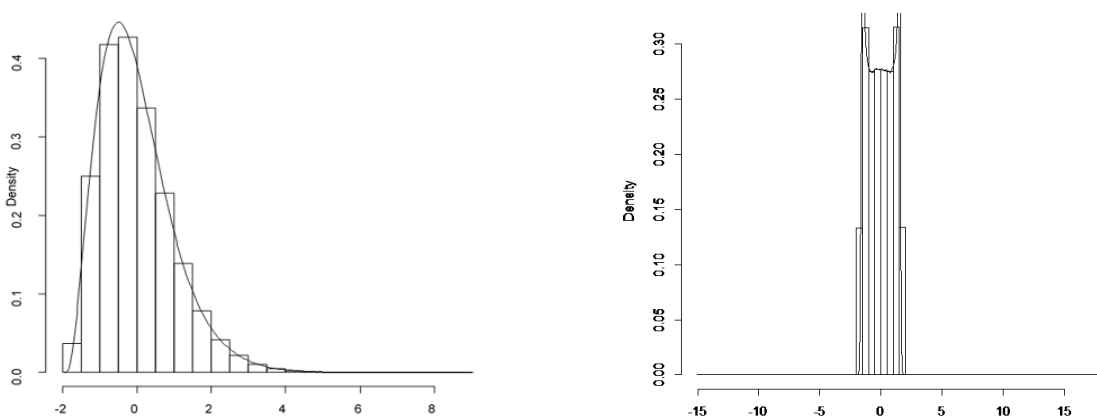


Figure 1: Simulated Densities of Fleishman Distributions with $(\gamma_1, \gamma_2) = (1, 1.5)$ and $(\gamma_1, \gamma_2) = (0, -0.5)$, respectively.

3. The Fleishman Vector Equation

It is well-known that

$$E(\mathbf{x}^{2k}) = \frac{(2k)!}{2^k k!}, E(\mathbf{x}^{2k+1}) = 0 \quad (k = 0, 1, \dots)$$

are the expectations of powers of \mathbf{x} . Because of

$$E(\mathbf{y}) = a + bE(\mathbf{x}) + cE(\mathbf{x}^2) + dE(\mathbf{x}^3) = 0$$

we arrive at the first equation

$$a + c = 0. \quad (\text{F0})$$

This leads by elimination of a to

$$\mathbf{y} = \mathbf{y}(b, c, d) = c(\mathbf{x}^2 - 1) + \mathbf{x}(b + d\mathbf{x}^2)$$

and by simple calculations to

$$E(\mathbf{y}^2) = b^2 + 2c^2 + 15d^2 + 6bd.$$

Observing $E(\mathbf{y}^2) = V(\mathbf{y}) = 1$ we get the second equation

$$f_1(b, c, d) := b^2 + 2c^2 + 15d^2 + 6bd - 1 = 0. \quad (\text{F1})$$

In principle we can substitute now c by a function of b and d and obtain in such a way the two-parametric $\mathbf{y} = \mathbf{y}(b, d)$. To avoid sophisticated expressions, we will further work with the three variables b , c and d . By using $\mathbf{y} = \mathbf{y}(b, c, d)$, calculating \mathbf{y}^2 , \mathbf{y}^3 and the corresponding expectations under the observation that the expectations of the odd powers of \mathbf{x} can be omitted we find the third and fourth equation

$$E(\mathbf{y}^3) = f_2(b, c, d) := 8c^3 + 6b^2c + 270cd^2 + 72bcd = \gamma_1 \quad (\text{F2})$$

$$E(\mathbf{y}^4) - 3 = f_3(b, c, d) := 3b^4 + 60c^4 + 10395d^4 + 60b^3d + 3780bd^3 + 60b^2c^2 + 630b^2d^2 + 4500c^2d^2 + 936bc^2d - 3 = \gamma_2 \quad (\text{F3})$$

Therefore, we have to consider the system of equations (F1), (F2) and (F3) to determine b , c and d . The coefficient a follows directly from (F0). Naturally, there are a lot of equivalent systems. Especially, (F1) can be used to reduce large coefficients in (F2), (F3) and to get a more transparent form of the system. Moreover, we introduce some convenient abbreviations. We get directly

$$f_1(b, c, d) = p + 2c^2 = 1, \quad p := b^2 + 15d^2 + 6bd$$

$$f_2(b, c, d) = 2cq = \gamma_1, \quad q := b^2 + 105d^2 + 24bd + 2$$

and using (F1) by adding multiples of this productive zero to (F3) finally

$$f_3(b, c, d) = 24(rc^2 + sd^2 + bd) = \gamma_2$$

$$r := b^2 + 141d^2 + 28bd + 1, \quad s := 225d^2 + 48bd + 12.$$

Since the quadratic terms p , q , r and s do not contain c , it can easily be separated and eliminated, respectively. This will be realized in the next sections. Besides, (F1) contains neither γ_1 nor γ_2 , while γ_1 occurs only in (F2) and γ_2 occurs only in (F3). These special features simplify the considerations. In compact form the algebraic equations (F1), (F2) and (F3) supply the *Fleishman vector equation* with *Fleishman vector function* $F: \mathbb{R}^3 \mapsto \mathbb{R}^3$ satisfying

$$F(b, c, d) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} (b, c, d) = \begin{pmatrix} 0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix}. \quad (\text{FE})$$

Obviously, the function F is smooth and maps bounded (closed) sets onto bounded (closed) sets. But it is not uniquely invertible.

In the following we are interested in the solution set

$$L = L(F, \gamma_1, \gamma_2) \subseteq \mathbb{R}^3$$

of (FE) for given γ_1, γ_2 and in the union $L_{\cup} := \bigcup_{(\gamma_1, \gamma_2)} L(F, \gamma_1, \gamma_2)$ of all such solutions sets. Observe that $L(F, \gamma_1, \gamma_2)$ is empty if only (complex) pseudo-solutions occur.

We denote by $R_{23}(F)$ the subset of the range

$$R(F) = \{(x, y, z)^T = F(b, c, d) : b, c, d \in \mathbb{R}\}$$

whose elements have the first coordinate $x = 0$, that is the part of $R(F)$ in the $(y-z)$ -plane. We will omit $x = 0$ in the following such that $R_{23}(F) \subseteq \mathbb{R}^2$. Besides, we look for pairs $(\gamma_1, \gamma_2) \in \Gamma := F(L_{\cup})$, which lead to solutions of (FE). Such pairs we call *admissible*. Since elements of L_{\cup} fulfil (F1) we have $F(L_{\cup}) \subseteq R_{23}(F)$. Later we will show that $F(L_{\cup}) = R_{23}(F)$.

An analysis shows that the algebraic system (FE) has the order 24. It can be reduced to a polynomial equation $P_{24}(d)=0$ of 24th degree in d (using e.g. the

symbolic toolbox of Matlab). Unfortunately, the polynomial expression is so long and non-transparent that we do not present it here. But, as a consequence, we can extend the domain of F to \mathbb{C}^3 and expect 24 (generally complex) solution triples (b, c, d) . The complex solutions we denote in this context as *pseudo-solutions*. Here, we look for the solution triples in \mathbb{R}^3 arising for admissible (γ_1, γ_2) . Considering symmetry properties of the system (FE), namely

$$F(-b, c, -d) = F(b, c, d), \quad F(b, c, d) = \begin{pmatrix} 0 \\ \gamma_1 \\ \gamma_2 \end{pmatrix} \Rightarrow F(b, -c, d) = \begin{pmatrix} 0 \\ -\gamma_1 \\ \gamma_2 \end{pmatrix},$$

we find

$$\begin{aligned} (b, c, d) \in L(F, \gamma_1, \gamma_2) &\Rightarrow (-b, c, -d) \in L(F, \gamma_1, \gamma_2), \\ (b, -c, d) &\in L(F, -\gamma_1, \gamma_2). \end{aligned}$$

Therefore, investigations can be restricted to $d \geq 0$ and $\gamma_1 \geq 0$.

4. Geometric Aspects

First we want to separate c in the Fleishman equations and consider the geometric properties of the corresponding surfaces in (b, d, c) -coordinate space. The solution set L with these rearranged triples is denoted by $L_{b,d,c}$. Sometimes we will also reflect the pairs (b, d) of solution triples. The set of all these pairs is denoted by $L_{b,d}$.

The equation $f_1 = 0$ represents an ellipsoid E obtained from the standard form by clockwise rotation in the (b, d) -plane with an angle of nearly 11.6° .

Obviously, the solutions (b, d, c) have to lie on E , that is $L_{b,d,c} \subseteq E$.

The first equation has the equivalent form

$$c^2 = \frac{1}{2} (1 - p). \tag{F1c}$$

which means $p \leq 1$ since c should be real. Since the intersection $p = 1$ of the ellipsoid with the (b, d) -plane is an ellipse $E_{b,d}$, we get real solutions for (b, d) only inside this ellipse. Since the midpoint $(0,0)$ of the ellipse supplies the global minimum 0 of $p = p(b, d)$, we have $p \geq 0$ and consequently $0 \leq p \leq 1$ inside.

The ellipse has the halve axes

$$\frac{1}{\sqrt{\lambda_1}} \approx 0.253, \quad \frac{1}{\sqrt{\lambda_2}} \approx 1.613, \quad \lambda_{1,2} = 8 \pm \sqrt{58}$$

such that it can be embedded in a rectangle $R_{b,d} = [-1.6, 1.6] \times [-0.4, 0.4]$.

Now we have $L_{b,d} \subset E_{b,d} \subset R_{b,d}$. The ellipsoid has also these halve axis and additionally $\frac{1}{\sqrt{2}}$ in direction of the c -axis. This and

$$c^2 \leq \frac{1}{2}.$$

follow from (F1c). The ellipsoid is contained in the following cuboid

$$E \subset R_{b,d,c} = R_{b,d} \times [-0.8, 0.8] = [-1.6, 1.6] \times [-0.4, 0.4] \times [-0.8, 0.8].$$

The equation $f_2 = 0$ has the equivalent form

$$c = \frac{1}{2} \cdot \frac{\gamma_1}{q}, \quad (\text{F2c})$$

where $q(0,0) = 2$ and $q(b,d) > 0$ in $E_{b,d}$. Therefore c has in $E_{b,d}$ the same sign as γ_1 . For $\gamma_1 \geq 0$ it follows $c \geq 0$.

The surface of $f_2 = 0$ in the (b, d, c) -space tends to 0 for $|(b, d)| \rightarrow +\infty$. It has a fold cutting out a closed curve on the surface of the ellipsoid E .

Figure 2 shows the two surfaces in the given box for $\gamma_1 = 1$.

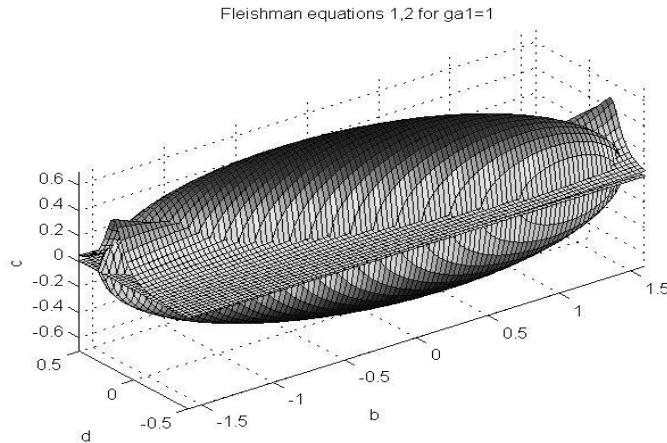


Figure 2: Surfaces for (F1) and (F2)

The equation $f_3 = 0$ can be transformed into the equivalent form

$$c^2 = \frac{\gamma_2 - 24(sd^2 + bd)}{24r}, \quad (\text{F3c})$$

where $r(0,0) = 1$ and $r(b,d) > 0$ in $E_{b,d}$. Hence, the denominator is strictly positive in $E_{b,d}$. For real c we get the inequality

$$24(sd^2 + bd) \leq \gamma_2.$$

The third surface connected with (F3c) has two parts ($c \geq 0, c < 0$). The upper part is shown in Figure 3 in the given box for $\gamma_2 = 2$.

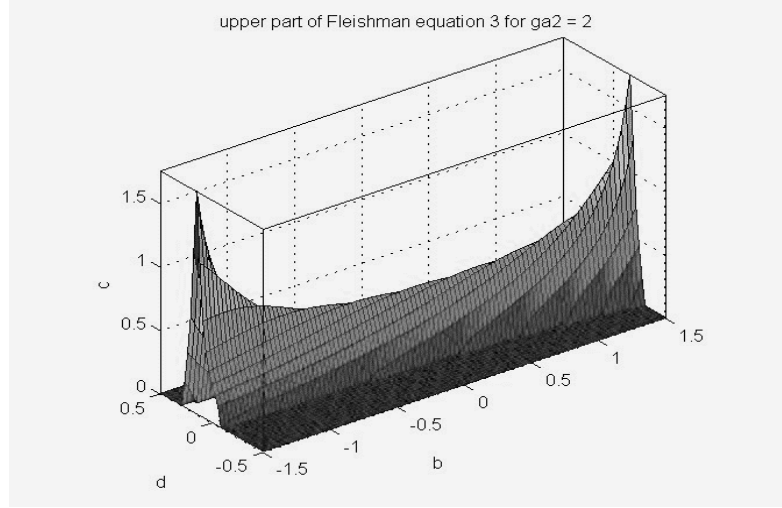


Figure 3: Surface for (F3)

As it is shown in section 5 we have 4 real solution triples in this case and therefore 4 intersection points of the three surfaces lying on E and in $R_{b,d,c}$. Hence, the absolute values of solutions b , c and d have to be small, especially that of d .

5. Solution Methods and first Results

As mentioned in section 3 it is practically impossible to solve the reduced polynomial equation $P_{24}(d) = 0$ of Fleishman equations (FE). Since (FE) can be written as a three-dimensional zero problem, iterative methods as the classical Newton method or some more sophisticated methods can be applied. This approach was also used in the original paper [1] of Fleishman. Disadvantages are that you need appropriate starting values and that you get single solutions. Without background knowledge about the number of solutions it is difficult to guarantee that all solutions are found. In the meantime, Computer Aided Systems

(CAS) as e.g. the Symbolic Toolbox of MATLAB are able to supply all solutions (with wanted precision) including the complex pseudo-solutions.

Considering the special equation system (FE) with $\gamma_1 = 1$ and $\gamma_2 = 2$, which is graphically described in Section 4 (Figures 2 and 3), the table of solutions (b, c, d) in [3] using results of [2] presents

$$s1: b = 0.9047 \dots, c = 0.1472 \dots, d = 0.0238 \dots$$

Considering this table one could think that the Fleishman equations (FE) have always one solution with positive values b, c and d . But true is that there are 24 solutions where some of them can be real but need not to be positive. MATLAB presents apart from the given solution triple (b, c, d) three further real ones, namely

$$s2: b = -0.9047 \dots, c = 0.1472 \dots, d = -0.0238 \dots,$$

$$s3: b = -1.2702 \dots, c = -0.3455 \dots, d = 0.1661 \dots,$$

$$s4: b = 1.2702 \dots, c = -0.3455 \dots, d = -0.1661 \dots$$

and 20 triples $s5$ to $s24$ with imaginary or complex values (pseudo-solutions).

Considering the real triples, that is the solutions, there are two pairs differing only in the signs. This is a consequence of the mentioned symmetry property of (FE). Therefore we have only two essentially different real solution triples which we can e.g. pick out by the restriction $d \geq 0$.

If we look at the 24 d -values in a compass plot we recognize 6 quadruples (see Figure 4 left). The quadruple structure is related to the reduced polynomial equation $P_{24}(d) = 0$ of (FE), mentioned before.

Having in mind the symmetry of (FE) with d also $-d$ is a solution. Besides, we know that with d and $-d$ also the conjugate complex \bar{d} and $-\bar{d}$ are solutions. In the real case the quadruple lies on the real line. Choosing another pair $(\gamma_1, \gamma_2) = (2.5, 9)$ we obtain 24 complex solution triples with no real value d , that is, no real solution triple. Again the quadruple structure of the d -values is visible (see Figure 4 right). It is worth to mention that both (γ_1, γ_2) -pairs fulfil Pearson's parabola inequality (2).

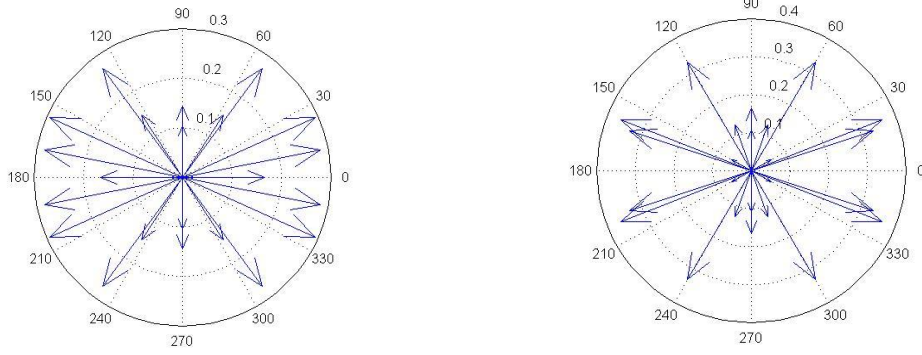


Figure 4: Geometry of d -values in the complex plane

A natural supposition is that different solution triples of (FE) have different higher moments. Hence, in the first step we calculate the fifth moment of y (in the standardized form). We omit the boring calculations and present one of the possible expressions:

$$\begin{aligned}
 E(y^5) = f_4(b, c, d) := & 604c^5 + 60b^4c + 623700cd^4 + 1800b^3cd \\
 & + 680b^2c^3 + 13440bc^3d + 189000bcd^3 \\
 & + 78600c^3d^2 + 25200b^2cd^2. \tag{F4}
 \end{aligned}$$

In the special case with $\gamma_1 = 1$ and $\gamma_2 = 2$ we get

$$f_4(s1) = f_4(s2) = 15.588 \dots, \quad f_4(s3) = f_4(s4) = -34.531 \dots$$

and see that the fifth moments of essentially different solution triples are indeed different.

6. The reduced Fleishman Equations

Now we want to eliminate c in the Fleishman vector equation. This is simple if we square (F2c) and replace there and in (F3c) the single term c^2 by its expression in (F1c). Thus we get the system

$$G(b, d) = \begin{pmatrix} g_1 \\ g_2 \end{pmatrix} (b, d) = \begin{pmatrix} \gamma_1^2 \\ \gamma_2 \end{pmatrix} \tag{RFE}$$

$$g_1(b, d) := 2(1-p)q^2, \quad g_2(b, d) := 12(1-p)r + 24(sd^2 + bd)$$

of two algebraic equations (RFE1) and (RFE2) with the two unknowns b and d . The pairs (b, d) of solution triples (b, c, d) have necessarily to fulfil these

equations. The question is if vice versa a solution pair (b, d) of (RFE) can be completed to a solution triple (b, c, d) of (FE) by an appropriate value c . The permuted triple (b, d, c) must lie on the ellipsoid E which means $0 \leq p \leq 1$. But $p \geq 0$ follows from the definition of p and $p \leq 1$ is a consequence of (RFE1). The value of c should be

$$c = \text{sign}(\gamma_1) \cdot \frac{1}{\sqrt{2}} \sqrt{1 - p}.$$

Then (F1c) is fulfilled. Observing that $q > 0$ for $0 \leq p \leq 1$ we can extract the root in (RFE1) and get (F2c). Finally, we can eliminate $\frac{1}{2}(1 - p)$ in (RFE2) and replace it by c^2 according the fixing of c which supplies (F3c). But these three equations are equivalent to (FE).

The solution of the reduced Fleishman equations is not easier for CAS than the solution of the Fleishman equations itself, but the geometrical representation supplies 2d-curves instead of 3d-surfaces. In Section 7 we will see that also the range of admissible pairs can be determined simpler.

We denote by $L_R = L_R(G, \gamma_1, \gamma_2)$ the solution set of (RFE) for given γ_1, γ_2 and by $L_{RU} := \bigcup_{(\gamma_1, \gamma_2)} L_R(G, \gamma_1, \gamma_2)$ the union of the solution sets for all admissible pairs (γ_1, γ_2) .

Studying different admissible pairs (γ_1, γ_2) two solution types can be identified. Restricting to real solutions with $d \geq 0$ we have for the two solution pairs either a sign change for b or not, namely

Type 1: $d_1 \geq 0, b_1 \geq 0; d_2 \geq 0, b_2 \leq 0$ (quadrants I, II),

Type 2: $d_1 \geq 0, b_1 \leq 0; d_2 \geq 0, b_2 \leq 0$ (quadrant II).

Besides, there are pairs (γ_1, γ_2) which are not admissible. The Figures 5 and 6 show in each case three curves, that for the ellipse $E_{b,d}$, that for (RFE1) and that for (RFE2). Figure 5 relates to admissible parameter pairs with solutions of type 1 ($\gamma_1 = 1, \gamma_2 = 2$) and type 2 ($\gamma_1 = 1, \gamma_2 = 50$). Figure 6 illustrates the case

where the parameter pairs are not admissible ($\gamma_1 = 1, \gamma_2 = -2$ and $\gamma_1 = 2.5, \gamma_2 = 4$, respectively). The next section investigates the range of admissible pairs in more detail.

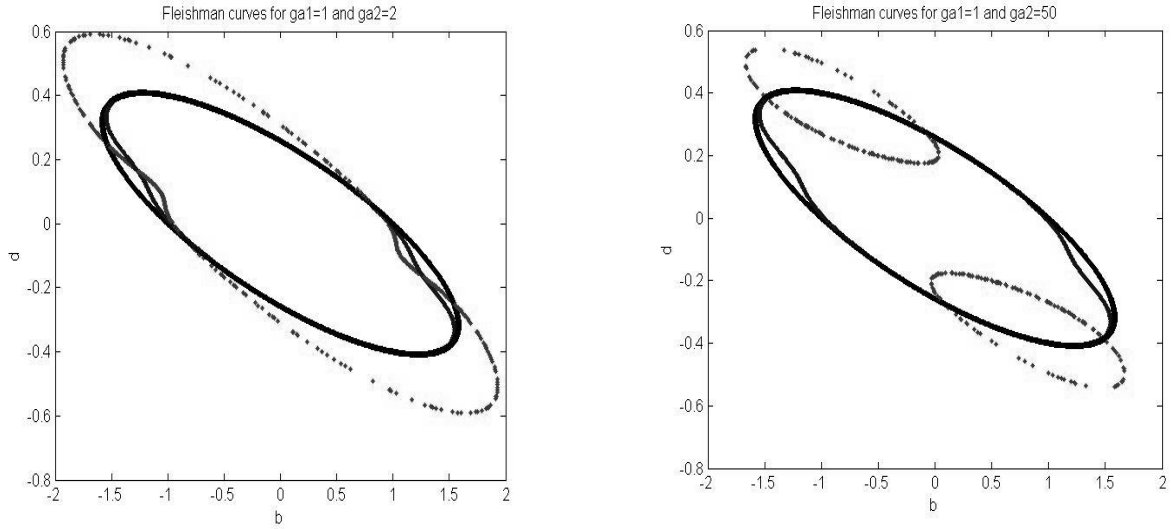


Figure 5: Curves for $E_{b,d}$ (bold), (RFE1) and (RFE2) (dotted) with admissible $\gamma_1 = 1, \gamma_2 = 2$ and $\gamma_1 = 1, \gamma_2 = 50$, respectively

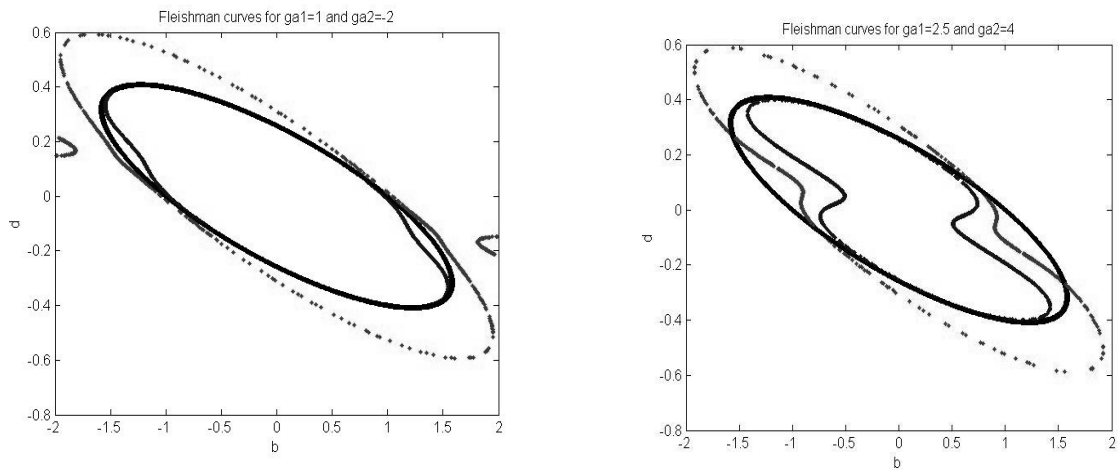


Figure 6: Curves for $E_{b,d}$ (bold), (RFE1) and (RFE2) (dotted) with not admissible $\gamma_1 = 1, \gamma_2 = -2$ and $\gamma_1 = 2.5, \gamma_2 = 4$, respectively

7. The Range of admissible Skewness and Excess Parameters

The vector function F in (FE) is in the context of Functional Analysis a finite-dimensional operator. Hence, it is continuous and compact which means that compact sets are mapped onto compact sets. All sets in the subset chain

$$L_U = \bigcup_{(\gamma_1, \gamma_2)} L(F, \gamma_1, \gamma_2) \subseteq E \subset R_{b,d,c},$$

namely the set L_U of all solutions of (FE), the ellipsoid E representing the first equation in (FE) and the cuboid $R_{b,d,c}$ containing E are compact. Hence, the set $\Gamma = F(L_U)$ of admissible pairs (γ_1, γ_2) for (FE) is therefore also compact. On the other hand, considering the connection of (FE) and (RFE) we will get Γ if we determine the range of G restricted to the convex hull $\text{conv}(E_{b,d})$ of $E_{b,d}$, that is

$$R(G|0 \leq p \leq 1) = R(g_1|0 \leq p \leq 1) \times R(g_2|0 \leq p \leq 1) = \left\{ \begin{pmatrix} \gamma_1^2 \\ \gamma_2 \end{pmatrix} \right\}$$

and deduce from them the admissible pairs. A simulation calculating the images of grid points in the rectangle $R_{b,d}$ belonging to $\text{conv}(E_{b,d})$ shows in Figure 7 a discrete version of the bounded region Γ_+ which contains all admissible pairs (γ_1, γ_2) with $\gamma_1 \geq 0$. This region can be completed by the mirrored part Γ_- of pairs (γ_1, γ_2) with $\gamma_1 \leq 0$ to get the whole admissible region $\Gamma = \Gamma_+ \cup \Gamma_-$.

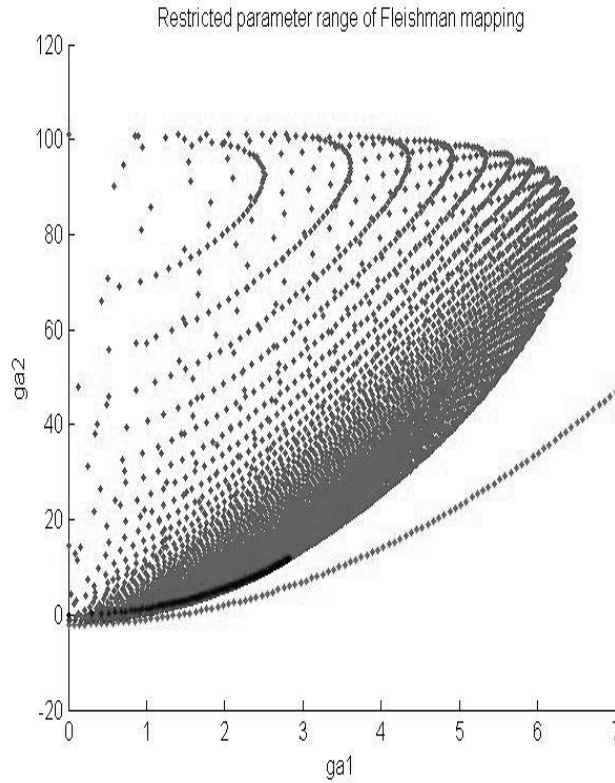


Figure 7: Right part of admissible range

The lower dark curve segment is explained in Section 8. The bottom curve starting in $(-2,0)$ is a part of the bounding parabola belonging to Pearson's inequality (2). Necessarily the points of admissible pairs lie above this parabola.

Finally, the discussion shows that

$$L_{R\cup} = \text{conv}(E_{b,d}), \quad L_{\cup} = \text{conv}(E).$$

8. The special case $d = 0$

Special cases can help to get some more insight in a complex problem. Here we restrict to $d = 0$ which leads to chi-quadrat distributions with one degree of freedom [1]. The Fleishman system (FE) degenerates to three equations

$$f_1(b, c) = b^2 + 2c^2 - 1 = 0 \quad (\text{FE1s})$$

$$f_2(b, c) = 2(b^2 + 2)c = \gamma_1 \quad (\text{FE2s})$$

$$f_3(b, c) = 24(b^2 + 1)c^2 = \gamma_2 \quad (\text{FE3s})$$

with only two unknowns b and c . The first equation represents an ellipse, the other two one-parametric curves. Since solution pairs (b, d) have to lie on this ellipse and $f_3(b, d) \geq 0$ we get the restrictions

$$b^2 \leq 1, \quad 2c^2 \leq 1, \quad \gamma_2 \geq 0.$$

The reduced Fleishman system (RFE) contains only the unknown b :

$$2(1 - b^2)(2 + b^2)^2 = \gamma_1^2 \quad (\text{RFE1s})$$

$$12(1 - b^2)(1 + b^2) = \gamma_2 \quad (\text{RFE2s})$$

From (RFE1s) we get $b^2 \leq 1$ and from (RFE2s) $0 \leq \gamma_2 \leq 12$. Obviously the two equations define a relation between γ_1 and γ_2 . This relation

$$\gamma_1^2 = 2 + \frac{1}{2} \gamma_2 - 2 \left(1 - \frac{1}{12} \gamma_2\right) \cdot \sqrt{1 - \frac{1}{12} \gamma_2}, \quad 0 \leq \gamma_2 \leq 12 \quad (3)$$

has to hold to get two solutions for b which follow from

$$b^2 = \sqrt{1 - \frac{1}{12} \gamma_2}$$

considering (RFE2s). Then we find $c = \frac{1}{2} \text{sign}(\gamma_1) \sqrt{1 - b^2}$ from (FE1s) and (FE2s). Hence, we get two solution triples $(b_1, c_1, 0)$ and $(b_2, c_2, 0)$. The relation (3) is given in Figure 7 by a dark curve segment.

Points (γ_1, γ_2) satisfying (3) are for example

$$(0,0), (\pm 2.5,9), (\pm 2\sqrt{2}, 12).$$

For $(\gamma_1, \gamma_2) = (2.5,9)$ there are 4 (real) solutions of solution type 1 for (FE) (see section 6), namely

$$s1: b = 0.7071 \dots, c = 0.5000 \quad , d = 0$$

$$s2: b = -0.7071 \dots, c = 0.5000 \quad , d = 0$$

$$s3: b = -0.5257 \dots, c = 0.6405 \dots , d = 0.0373 \dots$$

$$s4: b = 0.5257 \dots , c = 0.6405 \dots, d = -0.0373 \dots$$

If $d = 0$ is assumed, the two solutions s1 and s2 remain as predicted. The Figures 8 and 9 show two different cases for (FE) under the condition $d = 0$, namely the curves representing (FE1s), (FE2s) and (FE3s) for $(\gamma_1, \gamma_2) = (2.5, 9)$ and for $(\gamma_1, \gamma_2) = (1, 2)$, respectively. In the first case (Figure 8) the curves intersect according to the above given s1 and s2 in the two points $(b_1, c_1) = \left(\frac{1}{2}\sqrt{2}, \frac{1}{2}\right)$ and $(b_2, c_2) = \left(-\frac{1}{2}\sqrt{2}, \frac{1}{2}\right)$.

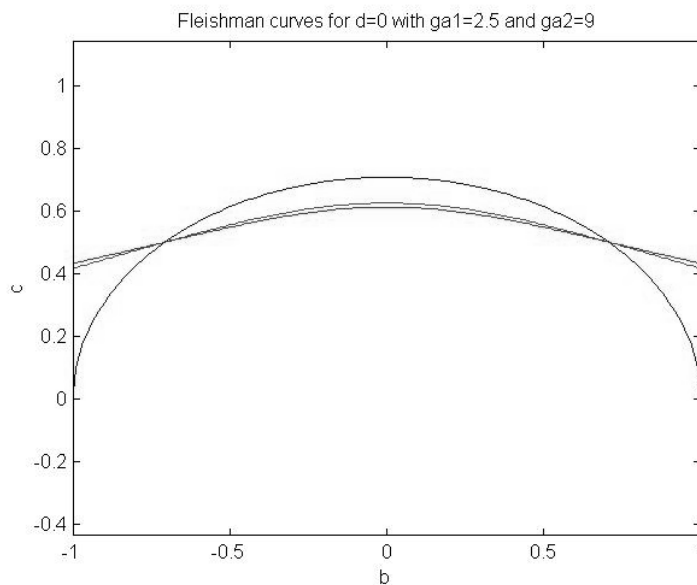


Figure 8: Curves for $d = 0, \gamma_1 = 2.5, \gamma_2 = 9$

In the second case (Figure 9) there is no common intersection point which means that for all solutions of (FE) we have $d \neq 0$ (see s1, s2, s3 and s4 in section 5).

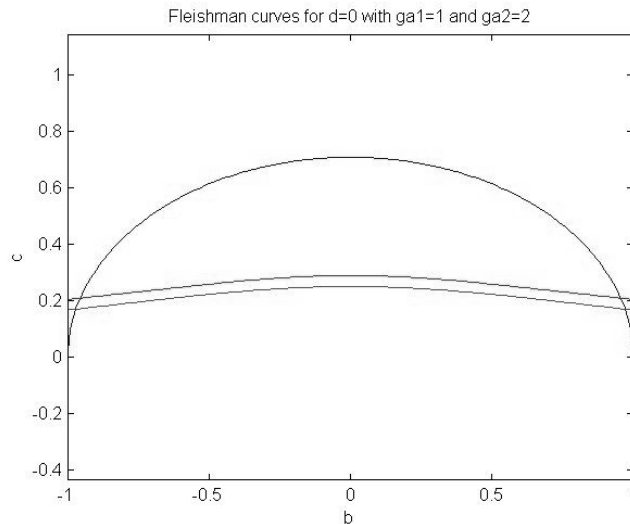


Figure 9: Curves for $d = 0, \gamma_1 = 1, \gamma_2 = 2$

References

1. Fleishman, A.J.: A method for simulating non-normal distributions. *Psychometrika* 43, 521-532 (1978).
2. Nürnberg, G.: Beiträge zur Versuchsplanung für die Schätzung von Varianzkomponenten und Robustheitsuntersuchungen zum Vergleich zweier Varianzen. *Probleme der angewandten Statistik*, Heft 6 (1982).
3. Rasch, D.: *Mathematische Statistik*, S. 217-218, Johann Ambrosius Barth Verlag, Leipzig 1995.
4. Rasch, D., Schott, D.: *Mathematical Statistics*, p. 453-454, Wiley 2018.
5. Schneider P., Verdooren, R., Rasch, D.: *Simulating Non-normal Distributions and the Definition of Robustness*, 10th International Workshop on Simulation and Statistics, Salzburg, September 2019.
6. Schott, D.: *The Calculation of the Coefficients of the Fleishman Distributions*, 10th International Workshop on Simulation and Statistics, Salzburg, September 2019.

Author

Prof. Dr. rer. nat. habil. Dieter Schott

Fakultät für Ingenieurwissenschaften, Gottlob-Frege-Zentrum

Hochschule Wismar

Philipp-Müller-Str. 14

D-23966 Wismar

E-Mail: dieter.schott@hs-wismar.de

Peter Junglas

Is body height really normally distributed?

Abstract. Analysing data sets of body heights one finds that for extensive amounts of data goodness-of-fit tests generally reject the hypothesis of normally distributed values. Numerical experiments quickly reveal that this is due to the rounding of the measured results. Further numerical studies show how this affects several tests and how one could cope with this problem. This can help to teach students how to utilise an “experimental” approach to mathematical questions. Furthermore it points out the difference between mathematical concepts like “real numbers” and actual data.

Introduction

Since statistics is applied almost everywhere, it is taught in many disciplines. The general approach usually is a mathematical one: Statistics is based on probability theory, its foundation is clear, its assertions and methods are proven.

Nevertheless statistics usually has a bad public reputation, that has not become better since Mark Twains dictum: “There are three kinds of lies: lies, damn lies, and statistics.” [1]. This is probably mainly due to flawed applications, results and statements made by people, who are better trained in the application area at hand than in mathematical intricacies. Common problems are wrong assumptions, such as data being normally distributed or independent, wrong conclusions, e. g. to mix up correlation and causality, or starting with wrong or biased data in the first place.

Therefore it is of paramount importance in teaching statistics to show not only correct methods, but to make students aware of the many pitfalls and errors that come up in everyday applications. The main purpose of this paper it to point out a special complication that is often overlooked and seldom taught, namely the consequences of using data that has been rounded. This will be done by presenting some simple numerical experiments that shed light on the problem. Hopefully this can motivate students to come up with experiments of their own to cope with the concrete challenges they will face in everyday statistical analysis.

Analysis of body heights

The analysis starts with two data sets of German female body heights containing $N = 300$ and $N = 3000$ entries. To get a first impression of their distribution, one can plot a simple histogram. This leads to the obvious assumption that the data is normally distributed, which can be tested qualitatively with a Q-Q plot. Fig. 1 shows the plots for $N = 300$, the other data set leads to similar results.

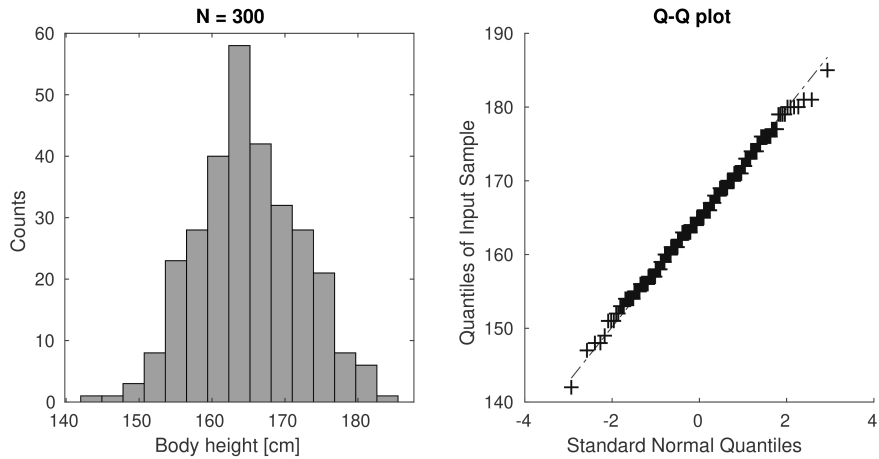


Figure 1: Histogram and Q-Q plot for $N = 300$.

A simple way to test the normality assumption is the well-known χ^2 goodness of fit test. To use it one has to partition the range of data values into non-overlapping intervals that span the whole range. One then counts the amount of data values in each interval (“bin”) and compares these numbers with the expected mean values under the hypothesis of normal distribution.

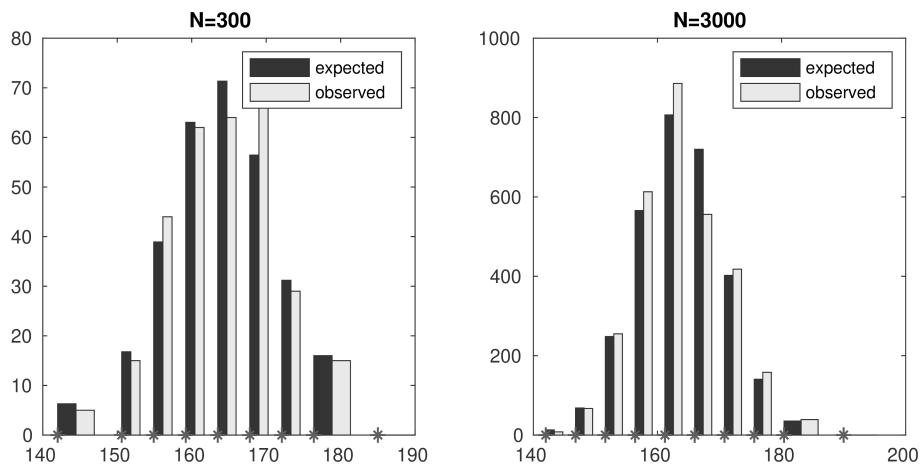


Figure 2: Expected and observed bin counts for both data sets.

The actual choice of intervals is not really important for the test as long as the following rules of thumb are fulfilled: All bins are non-empty and at least 80% of the bins contain more than 5 data elements [2]. For concreteness in the following we will use Matlabs function `chi2gof` that chooses appropriate intervals automatically. Corresponding results are shown in Fig. 2.

These results are now combined in a variable T that quantifies the difference between expected and observed values. From it one computes the p-value p of the test, i. e. the probability that normally distributed data values have a T value at least as large as the observed one. For the two data sets one gets

$$\begin{aligned} p(300) &= 0.470 \\ p(3000) &= 0.266 \cdot 10^{-3} \end{aligned}$$

While the first result is in good agreement with the assumption of normally distributed data, the second one definitely is not. For any reasonable confidence level one would reject the hypothesis. This is in contrast to common expectation about body heights and to the qualitative results of the histogram and Q-Q plot.

After some carefully guided discussion in class, students can come up with the suggestion that the rounding of the numbers to integer centimeter values might be the reason for the strange result. To validate this idea, one can perform a simple numerical experiment: One uses a random number generator to create normally distributed values similar to the given data (adopting $\mu = 165$, $\sigma = 6.9$), rounds them to integers and applies the χ^2 test. This basically reproduces the p-values from above, which confirms the significance of the rounding effect. In hindsight this seems obvious: Rounded values always come from a discrete distribution, which in our case we assume to be a rounded version of the normal distribution.

For small data sets the χ^2 test cannot discriminate between rounded and unrounded values, but with growing data size N the p-value decreases. To study this behaviour in more detail we repeat our experiment with varying N . Since the p-values largely fluctuate, each experiment is repeated three times and the mean values are plotted over N . They decrease drastically, therefore a semilogarithmic plot is used that hints at an exponential decrease of p (Fig. 3).

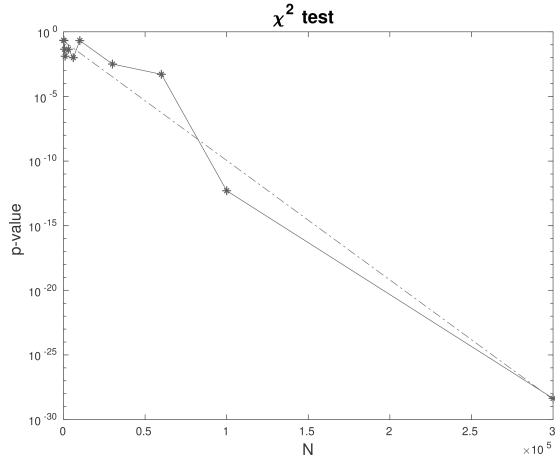


Figure 3: p-values using the χ^2 test.

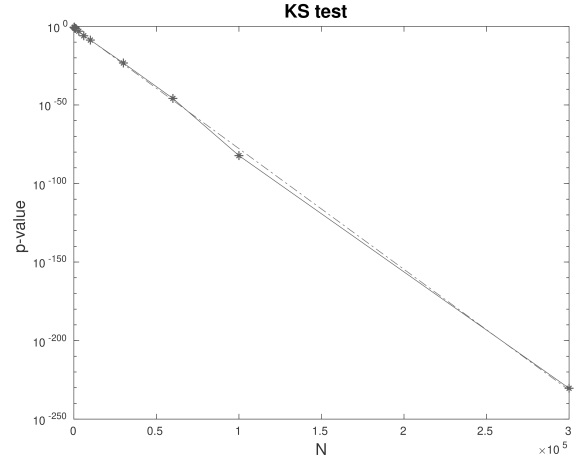


Figure 4: p-values using the KS test.

With the freedom to choose the bin sizes and positions, the χ^2 test has a lot of parameters, which complicates the analysis. Therefore the Kolmogorov-Smirnov test (“KS test”) is used to check our findings. Without any parameters it directly compares the empirical and given distribution functions and leads to a complicated, but well known test statistics. The results in Fig. 4 show that the p-values again decrease exponentially with N , but much faster than before. Apparently the KS test discriminates better between continuous and rounded values.

Choosing the bins in the χ^2 test

Since the χ^2 test is very common in many applications, we will now study, how the sizes and positions of the bins influence the ability to discriminate between exact and rounded values. We create rounded normal values as before and apply the χ^2 test, but this time we define the bins ourselves instead of relying on the choice of the Matlab routine.

We begin by using bins of identical width w , positioned such that one bin is centered around the mean value, and vary w . The resulting p-values as function of w are plotted for two data sizes in Fig. 5. One finds that the p-values are much larger at integer widths; this effect is more pronounced for odd width and for large N .

Next we fix the bin width at $w = 5$ and vary the position of the bins. Initially their edges are at integer values, than we shift them by a certain amount s and plot the p-values over s (Fig. 6). They are now minimal at integer, maximal at half-integer shifts.

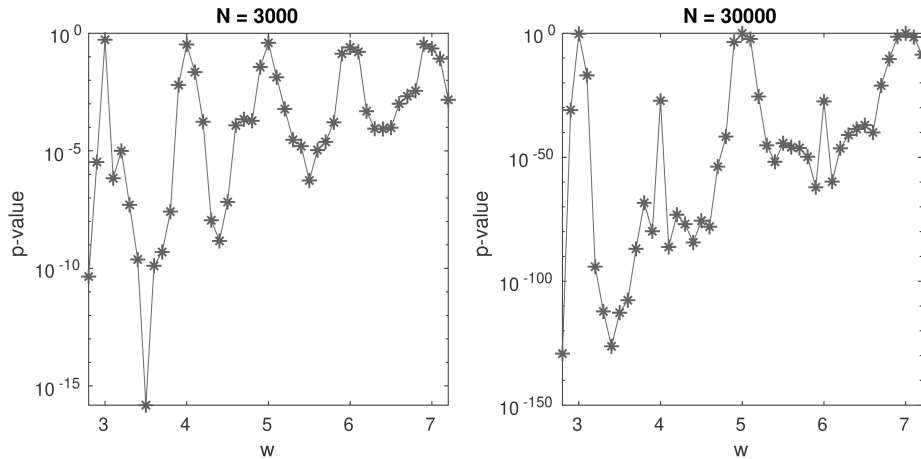


Figure 5: p-values for different bin widths.

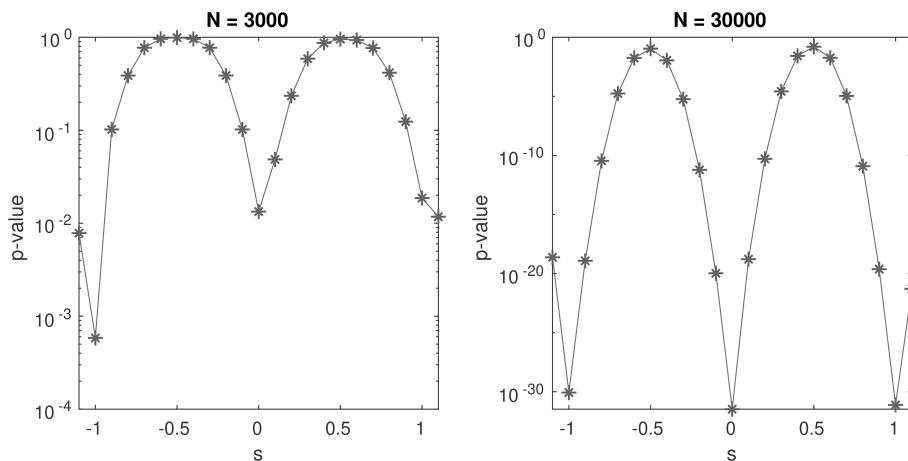


Figure 6: p-values for different bin positions.

Again, looking at the results, their explanation is easy: The p-values are maximal for bins of the type $[n_1 + \frac{1}{2}, n_2 + \frac{1}{2}]$ with $n_1, n_2 \in \mathbb{N}$. These bins are invariant under rounding, i. e. the expected bin counts with or without rounding are the same. In these cases the χ^2 test can't detect rounding and confirms the normal distribution of the (original) values. This as well explains the minimality of the odd integer widths: Since the mean value is $\mu = 169$, the corresponding boundaries of the centered bins lie on half-integer values.

In summary we suggest the following approach: If one wants to detect the difference between exact and rounded values, one should use bin boundaries at integers – or choose the KS test. If one is only interested in the distribution of the (theoretical) real values and wants to get rid of rounding effects, one should use bin boundaries at half-integers.

Varying precision

An interesting question is, how many data points one needs to detect the rounding for given measurement precision. For definiteness we will use a confidence level of $\alpha = 0.01$, choose a precision and experimentally determine the needed size N at which the p-value is smaller than 0.01.

The experiment starts by creating normally distributed numbers, but the rounding is now done using a scale w :

$$x_{\text{round}} := w \text{round}(x/w)$$

Here smaller values of w mean higher measurement precision. We use the KS test to measure the p-value as a function of the data size N for a set of fixed scale values w . The results are shown in Fig. 7, where again the p-values shown are mean values of three runs.

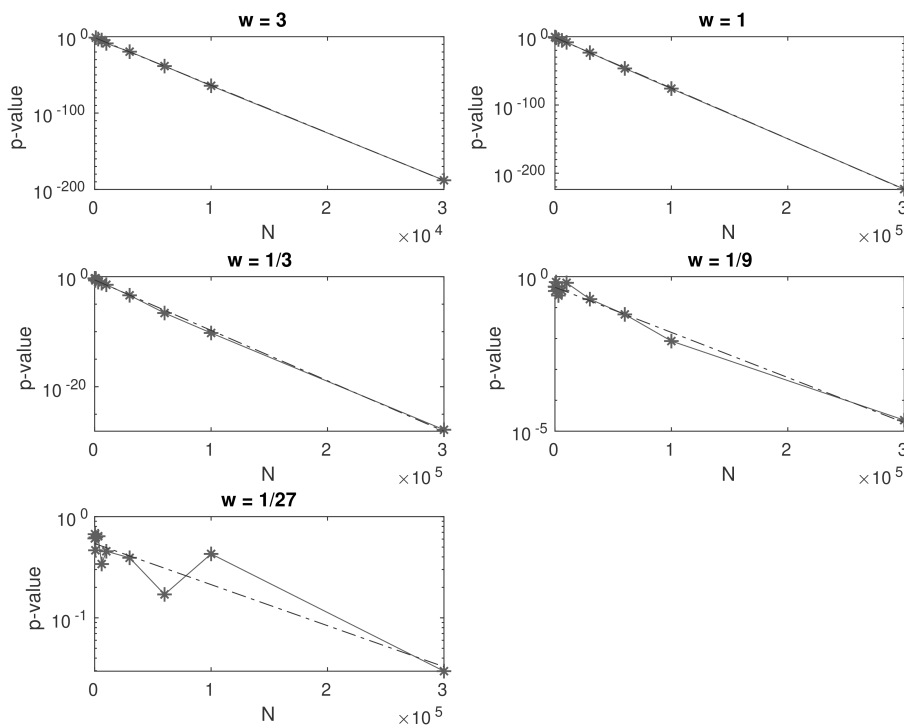


Figure 7: p-values over N for varying scale w .

Once more we can see the exponential decline, i. e. approximately we have

$$\log(p) = aN + b$$

The parameters a , b of corresponding fitting lines are computed and the

lines shown in the figure. We now use them to compute the size N_0 , at which the p-value reaches the confidence level α and get

w	N_0	$w^2 \cdot N_0$
3.00	122	1098
1.00	1279	1279
0.33	15358	1706
0.11	113500	1401
0.04	427332	586

The last row is inaccurate, since the data size N used in the experiments was not large enough. Considering this and the large scale of values, the third column convincingly reveals the relationship

$$N_0 \sim \frac{1}{w^2}.$$

Therefore increasing the precision by a factor 10, one needs 100 times as much data to see the rounding effect in a goodness-of-fit test.

We have only used a fixed data distribution, especially a fixed standard variation σ . It is well known that the relevant parameter describing the importance of rounding is σ/w [3]. This can be seen experimentally or by a simple scaling argument.

Smoothing the distribution

A completely different way to look at rounding comes from signal theory, where it is called “quantization” [4]. The basic idea is to look at the effects of rounding as a kind of noise. One can prove that under certain conditions this noise has special properties, which might allow to replace the actual discrete distribution function of the rounded values by a continuous one. This makes its further use in statistical computations much simpler. The necessary conditions are not fulfilled exactly in our case, but at least approximately.

We begin by writing

$$X_s = X + Y$$

where X describes our (original) data which is normally distributed,

$$X \sim N(\mu, \sigma^2),$$

and X_s is the rounded data,

$$X_s = \text{round}(X).$$

The difference Y is now interpreted as noise, which has approximately a uniform distribution: $Y \sim U(-0.5, 0.5)$. Of course X and Y are not independent, but they are at least approximately uncorrelated.

To check whether these approximations are valid in our example, we test them experimentally. We create normally distributed random values X , round them to get X_s and compute $Y = X - X_s$. We then test for the distribution of Y using a χ^2 test, compute the correlation coefficients and get

N	p-value	$\rho(X, Y)$
3000	0.6165	-0.0077
30000	0.5279	0.0085
300000	0.4733	0.0003

So the assumptions about Y seem to be reasonable here. If we further assume that X and Y are approximately independent, we can use their known distribution functions to compute the density function f_s of X_s via convolution and get

$$f_s(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} dz e^{-\frac{(x-z-\mu)^2}{2\sigma^2}}$$

This is the standard Gaussian function smoothed over an interval of width one, which coincides with the probabilities of X_s at integer x . Of course it can be easily computed numerically, as well as the corresponding cumulative distribution function F_s . For the values of our standard example f_s is shown in Fig. 8.

After all this work it's time to try out our new density function and test, whether it really is a good approximation for the distribution of rounded values. Therefore we once again create rounded random numbers and perform a χ^2 test, this time using f_s instead of the normal distribution function. We use edges on integral boundaries to make sure that the test is sensitive to rounding, and get

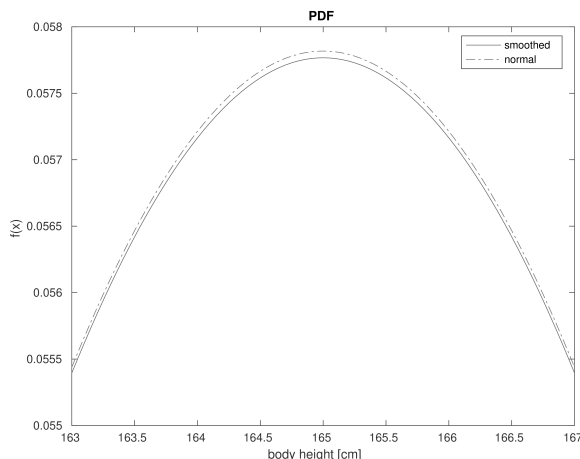


Figure 8: Normal and smoothed normal density functions.

N	p-value (normal)	p-value (f_s)
3000	0.0044	0.0046
30000	3.7098e-42	4.9609e-42
300000	0	0

The “smoothed” distribution function is as bad as the original one. Apparently the whole idea is completely useless for a goodness-of-fit test.

Conclusions

Though it is long known that rounding can have an effect on statistical evaluations, research on precise ways to cope with it is rather recent. Several rules of thumb existed that have been shown to be wrong in [3] and replaced by better ones. Using maximum-likelihood methods one can find correct statistical estimates of rounded data, e. g. confidence intervals, especially for small data sets and small values of σ/w [5].

Our experiments have shown that rounding can have an effect even for large data sets, at least for the results of a goodness-of-fit test. If one is interested in the distribution of the “real” (i.e. unrounded) values, one can use qualitative tools like a Q-Q plot or use adapted methods like a χ^2 test with properly chosen intervals.

Furthermore there are a few general lessons that students can learn from this study: First of all numerical experiments are a valuable tool to find facts, test ideas and identify relevant parameters. Second, statistical analysis is not always done best with a huge amount of data – sometimes this might lead to fine grained details one is not interested in. Third,

“reasonable” assumptions or approximations might be wrong, therefore one has always to question everything – even if it has been used for decades.

Finally we have to answer the initial question: Is body height really normally distributed? The immediate answer “no, since it cannot be negative” is irrelevant, because the probability of negative heights is much too small in our example to be of any consequences. That a body height has to be measured and therefore has always rounded values, is more relevant for our question, but on the other hand one can usually increase the precision, until this point doesn’t really matter. The true answer is, that a “body height” doesn’t exist! What could it possibly refer to: the position of the “highest” electron of your body? According to Heisenberg’s uncertainty principle this concept makes no sense.

One has always to keep in mind that “real numbers” are a mathematical abstraction, while in statistics one is generally talking about the real world. Whether a mathematical model fits the situation at hand, is generally an important question, not only because of rounding.

Bibliography

- [1] **Twain, M.:** *Autobiography Vol. I.* Ed. A. B. Paine, Harper Brothers New York and London (1924).
- [2] **Ross, S. M.:** *Introduction to Probability and Statistics for Engineers and Scientists.* Academic Press London, 5th ed. (2014).
- [3] **Tricker, A.; Coates, E.; Okell, E.:** *The Effect on the R Chart of Precision of Measurement.* J. Qual. Technol., 3, **30**, 232–239 (1998).
- [4] **Widrow, B.; Kollar, I.; Liu, M.-C.:** *Statistical theory of quantization.* IEEE Trans. Instrum. Meas., 2, **45**, 353–361 (1996).
- [5] **Vardeman, S. B.; Lee, C.-S.:** *Likelihood-based statistical estimation from quantized data.* IEEE Trans. Instrum. Meas., 1, **54**, 409–414 (2005).

Author

Prof. Dr. rer. nat. Peter Junglas
 Private Hochschule für Wirtschaft und Technik Vechta/Diepholz
 Schlesierstraße 13a
 D-49356 Diepholz
 E-Mail: peter@peter-junglas.de

Anhang

WFR - Wismarer Frege-Reihe / Wismar Frege Series

Beiträge zur Mathematikausbildung von Ingenieuren

- Heft 01/2005 Proceedings 4. Workshop Mathematik für Ingenieure, Bremen, Oktober 2005.
- Heft 05/2006 Proceedings 5. Workshop Mathematik für Ingenieure, Wismar, Teile 1 – 3, September 2006.
- Heft 01/2007 Proceedings Minisymposium Moderne Mathematikausbildung für Ingenieure, Humboldt-Universität Berlin, Teile 1 – 2, März 2007.
- Heft 02/2007 Mathematik für Ingenieure – Thesen zum Jahr der Mathematik 2008, Dezember 2007.
Mathematics for Engineers – Theses to the Year of Mathematics 2008, December 2007.
- Heft 03/2008 Proceedings 6. Workshop Mathematik für Ingenieure, Soest, Teile 1 – 2, September 2008.
- Heft 04/2008 Proceedings Minisymposium Moderne Mathematikausbildung für Ingenieure, Friedrich-Alexander-Universität Erlangen-Nürnberg, Teile 1 – 2, September 2008.
- Heft 03/2009 Peter Junglas: Interaktive Simulationsprogramme zur Demonstration von klassischen und quantentheoretischen Wellenphänomenen, Juni 2009.
- Heft 04/2009 Proceedings 7. Workshop Mathematik für Ingenieure, Wolfenbüttel, Juni 2009.
- Heft 02/2010 Information – Programme and Abstracts, 15th SEFI MWG Seminar & 8th Workshop GFC, Wismar, June 2010.
- Heft 03/2010 Proceedings 8. Workshop Mathematik für Ingenieure, Wismar, Juni 2010.
- Heft 05/2010 Larissa Fradkin: Teaching Algebra and Calculus to Engineering Entrants, December 2010.
- Heft 02/2011 Proceedings 9. Workshop Mathematik für ingenieurwissenschaftliche Studiengänge, Wilhelmshaven, September 2011.
- Heft 03/2013 Proceedings 11. Workshop Mathematik für ingenieurwissenschaftliche Studiengänge, Teile 1 – 2, Bochum, September 2013.

Heft 02/2015	Proceedings 12. Workshop Mathematik für Ingenieure, Teile 1 – 2, Hamburg, Februar 2015.
Heft 04/2016	Proceedings 13. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen, Lingen, September 2016.
Heft 01/2017	Proceedings 14. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen, Erlangen, September 2017.
Heft 01/2018	Sergiy Klymchuk: Puzzle-Based Learning in Engineering Mathematics: Students' Attitudes.
Heft 02/2018	Proceedings 1st Northern-Light Symposium on Engineering Education, Hamburg, April 2018.
Heft 02/2019	Proceedings 15. Workshop Mathematik in ingenieurwissenschaftlichen Studiengängen, Rostock-Warnemünde, April 2019.
Heft 03/2019	Proceedings 2nd Northern-Light Symposium on Mathematical Education in Engineering, Hamburg, September 2019.

Hinweis:

Die Proceedings zur Workshopreihe beginnen erst mit dem 4. Workshop.

Die Proceedings zum 10. Workshop Mathematik erschienen in einem Extraband an der Hochschule Ruhr/West in Mülheim.

Herausgeber und Redakteur

Prof. Dr. rer. nat. habil. Dieter Schott
Gottlob-Frege-Zentrum
Fakultät für Ingenieurwissenschaften
Hochschule Wismar
Philipp-Müller-Str. 14
D - 23966 Wismar
Telefon: ++49 / (0)3841 / 753 7333
Fax: ++49 / (0)3841 / 753 7130
E-Mail: dieter.schott@hs-wismar.de

Vertrieb:

Direkt über den Herausgeber oder das Gottlob-Frege-Zentrum

