

**THE DERIVE - NEWSLETTER #124**

**ISSN 1990-7079**

**THE BULLETIN OF THE**



**USER GROUP**

**+ CAS-TI**

**C o n t e n t s :**

- |    |  |
|----|--|
| 1  | Letter of the Editor                                       |
| 2  | Editorial - Preview  |
|    | Felipe de Jesús Martínez Vargas                            |
| 3  | Number Systems   |
|    | Wolfgang Alvermann & Josef Böhm                            |
| 5  | The Rosette  |
|    | Josef Böhm   |
| 7  | Double Die Encryption                                      |
|    | Michel Beaudin   |
| 15 | Résumé 1 - Part 3  |
|    | Bhuvanesh Bhatt & Josef Böhm & F. de Jesús Martínez Vargas |
| 24 | A Survey over Distributions 3                              |
| 45 | The Beta Function  |
| 48 | Greetings and Wishes                                       |

---

**December 2021**

|         |                                 |         |
|---------|---------------------------------|---------|
| DNL 124 | Information - Recommended Links | DNL 124 |
|---------|---------------------------------|---------|

Modellieren mit Mathematik (German only): <https://www.blick.it/blick/angebote/modellmathe/>

Activités BBC micro:bit avec la  
TI-83 Premium CE Edition Python

Apply Python for programming  
BBC micro:bit (in French)

Enseignement de la physique-chimie au lycée

<https://online.flipbuilder.com/wera/llxz/>

You can download a printed version (and other materials) from:

<https://education.ti.com/-/media/ti/files/france/downloads/pdf/activites-bbc-micro-bit-2.pdf?rev=3dcdfa28-1fa6-4c8f-8b13-aeefc642499b&la=fr&hash=50B9FDE818AEE6F46F3D996FD1B0FC70C20D8179>

<https://resources.t3france.fr/t3france>

<https://resources.t3france.fr/t3france>

Here you can find a lot of materials from the *innomath project*:

<https://innomath.eu/>



Erasmus+

## MATERIAL FOR GIFTED STUDENTS

B.1 A COLLECTION OF MATHEMATICS PROBLEMS FOR GIFTED/TALENTED SCHOOL STUDENTS (age under 14)

B.2 A COLLECTION OF MATHEMATICS PROBLEMS FOR GIFTED/TALENTED SCHOOL STUDENTS (age over 14)

B.3 PROJECTS AND APPLICATIONS

B.4. COMPETITIONS/CHALLENGES AND COMMUNICATION ACTIVITIES

B5. VIDEOS AND INTERACTIVE MEDIA RELATED TO GIFTEDNESS

## LEARNING PLANS FOR SUPPORTING GIFTED STUDENTS

Grades 5-8

Grades 9-12

## INNOMATH TRAINING COURSE FOR TEACHER FACILITATORS

Dear DUG-Members,

our lockdown also has a positive side, too (but one I would gladly do without). The DNL will be ready in time before Christmas.

In this issue, two contributions are continued: Michel Beaudin's *Résumé* and the *Survey over Probability Distributions*. I provide the translation of the French contribution into German and English again. Regarding the distributions, I recommend that *DERIVIAN*S have a look at the PDF Y CDF\_engl.dfw file contributed by Felipe de Jesús Martínez Vargas (Mexico).

I received more contributions from Mexico again, which I will present in later DNLs.

I came across the double cube cipher while reading a book. I was immediately fascinated by it because it can be done so easily with paper and pencil and has actually been used extensively in secret service.

In my experience, cryptography is always very interesting for the students. For me the programming was an exciting task, although I am sure that there could be a shorter program code. This is perhaps a task for the programmers among you.

One mail seems to be out of the ordinary. On page 14 you will find information from a professor in Tunisia. His papers can be downloaded from the website of the renowned forum for academic publications, *Researchgate*.

I also want to take this opportunity to point out the links on the first page. The *Innomath Project* website offers a lot of materials.

Finally, I want to thank our friend Alvermann. He inspires me again and again to new mathematical deeds and adventures.

With best wishes for Christmas and for the New Year 2022 we remain as always

Yours Noor and Josef

Liebe DUG-Mitglieder,

unser Lockdown hat auch eine positive Seite (auf die ich aber gerne verzichten würde). Der DNL wird rechtzeitig vor Weihnachten fertig.

In dieser Ausgabe werden zwei Beiträge fortgesetzt: Michel Beaudin's *Résumé* und die *Übersicht über die Wahrscheinlichkeitsverteilungen*. Die Übersetzung des französischen Beitrags ins Deutsche und Englische stelle ich wieder zur Verfügung. Zu den Verteilungen ist zu sagen, dass ich den *DERIVIANERN* empfehle, sich die Datei PDF Y CDF\_engl.dfw von F. Martínez Vargas (Mexiko) anzusehen.

Von Mexiko habe ich weitere Beiträge erhalten, die in späteren DNLs vorgestellt werden.

Auf die Doppelwürfel-Verschlüsselung bin ich beim Lesen eines Buches gestoßen. Sie hat mich gleich fasziniert, weil sie so einfach mit Papier und Bleistift ausgeführt werden kann und tatsächlich in großem Maß im Geheimdienst eingesetzt worden ist.

Nach meiner Erfahrung ist die Kryptografie immer sehr interessant für die Schüler und für mich war die Programmierung eine spannende Aufgabe, wobei ich sicher bin, dass es einen kürzeren Programmcode geben könnte. Das ist vielleicht eine Aufgabe für die Programmierer unter Euch.

Eine Mail fällt wohl aus unserem gewohnten Rahmen. Auf Seite 14 findet Ihr eine Information eines Professors aus Tunesien. Seine Aufsätze sind im renommierten Forum für akademische Publikationen, *Researchgate*, zu finden.

Bei dieser Gelegenheit will ich auch auf die Links auf der ersten Seite hinweisen. *Innomath* bietet eine Menge Materialien an.

Schließlich will ich mich noch bei unserem Freund Alvermann bedanken. Er inspiriert mich immer wieder zu neuen mathematischen Taten.

Mit besten Wünschen für Weihnachten und für das Neue Jahr 2022 verbleiben wir wie immer

Eure Noor und Josef



The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE & CAS-TI User Group*. It is published at least four times a year with a content of 40 pages minimum. The goals of the *DNL* are to enable the exchange of experiences made with *DERIVE*, *TI-CAS* and other CAS as well to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

Editor: Mag. Josef Böhm  
 D'Lust 1, A-3042 Würmla, Austria  
 Phone: ++43-(0)660 31 36 365  
 e-mail: nojo.boehm@pgv.at

### Contributions:

Please send all contributions to the Editor. Non-English speakers are encouraged to write their contributions in English to reinforce the international touch of the *DNL*. It must be said, though, that non-English articles will be warmly welcomed nonetheless. Your contributions will be edited but not assessed. By submitting articles, the author gives his consent for reprinting it in the *DNL*. The more contributions you will send, the more lively and richer in contents the *DERIVE & CAS-TI Newsletter* will be.

Next issue:

March 2022

### **Preview: Contributions waiting to be published**

Some simulations of Random Experiments, J. Böhm, AUT, Lorenz Kopp, GER  
 Wonderful World of Pedal Curves, J. Böhm, AUT  
 Simulating a Graphing Calculator in *DERIVE*, J. Böhm, AUT  
 Cubics, Quartics – Interesting features, T. Koller & J. Böhm, AUT  
 Logos of Companies as an Inspiration for Math Teaching  
 Exciting Surfaces in the FAZ, BooleanPlots.mth, P. Schofield, UK  
 Old traditional examples for a CAS – What's new? J. Böhm, AUT  
 Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZ  
 Tutorials for the NSpireCAS, G. Herweyers, BEL  
 Dirac Algebra, Clifford Algebra, Vector-Matrix-Extension, D. R. Lunsford, USA  
 Another Approach to Taylor Series, D. Oertel, GER  
 Charge in a Magnetic Field, H. Ludwig, GER  
 More Applications of TI-Innovator™ Hub and TI-Innovator™ Rover  
 Surfaces and their Duals, Cayley Symmetroid, J. Böhm, AUT  
 A Collection of Special Problems, W. Alvermann, GER  
 DERIVE Bugs? D. Welz, GER  
 Tweening & Morphing with TI-NspireCX-II-T, J. Böhm, AUT  
 The Gap between Poor and Rich, J. Böhm, AUT  
 More functions from M. Myers and from Bhuvanesh's Mathtools-library  
 TaxiCab Conics, Two alternate Approaches to Conics, R. Haas, USA  
 QR-Code light, Random numbers following a given distribution  
 Polygons and their properties, Quartiles, F. de Jesús Martínez Vargas, Mexico  
 Penalty shootout mathematics, B. Grabinger, GER  
 Sparse Matrices, 153 is another Special Number, and others

Impressum:  
 Medieninhaber: *DERIVE* User Group, A-3042 Würmla, D'Lust 1, AUSTRIA  
 Richtung: Fachzeitschrift  
 Herausgeber: Mag. Josef Böhm

# Number Systems

Felipe de Jesús Martínez Vargas, Mexico

Dear Josef Böhm,

It is a pleasure and a satisfaction to be able to contribute to your prestigious magazine.

Mexico is a wonderful, multicultural country, its great variety of food and music and especially its cheerful people, as you mention in your visit to our country. I live in Manzanillo, Colima; a place of tropical jungle and bathed by the Pacific Ocean.

Modify the number base conversion functions which I present to you:

```
CHARACTER := "0123456789ABCDEFGHIJKLMNPOQRSTUVWXYZ"
```

"Función conv\_BASE(x,b) convierte un número entero decimal a cualquier base"

```
conv_BASE(x, b, c_, d_, e_, f_, r_) := PROG(c_ := REVERSE_VECTOR(VECTOR(CHARACTER↓(MOD((ITERATES(FLOOR(i, b), i, x))↓k, b) + 1), k, 1, DIM(ITERATES(FLOOR(i, b), i, x)) - 2)), d_ := NAME_TO_CODES(c_)↓1, e_ := IF(c_ = [], CHARACTER↓b, APPEND("000000"↓[1, ..., DIM(c_) - DIM(CODES_TO_NAME(d_))], CODES_TO_NAME(d_))), f_ := CODES_TO_NAME(VECTOR(NAME_TO_CODES(CHARACTER↓b), i, 1, 10)↓1), IF(SIGN(x) = -1, r_ := f_↓[1, ..., 10 - DIM(e_)], r_ := ""), IF(x = 0, "0", APPEND("", r_, e_)))
```

"Función conv\_DEC(x,b) convierte un número de cualquier base a entero decimal"

```
conv_DEC(x, b, r_, s_) := PROG(IF(DIM(x) = 10 ∧ x↓1 = CHARACTER↓b, PROG(s_ := -1, x := DELETE(x, 1), r_ := REVERSE(VECTOR(POSITION(x↓i, REVERSE(CHARACTER↓[1, ..., b]))) - 1, i, DIM(x)))), PROG(s_ := 1, r_ := REVERSE(VECTOR(POSITION(x↓i, CHARACTER↓[1, ..., b]) - 1, i, DIM(x))))), s_·Σ(r_↓i·b^(i - 1), i, 1, DIM(r_)) + IF(s_ = 1, 0, -1))
```

Examples :

conv\_BASE(258, 16) = 102

conv\_BASE(27, 2) = 11011

conv\_BASE(-258, 16) = FFFFFFFEFE

conv\_BASE(1199, 12) = 83B

conv\_DEC(FFFFFFEFE, 16) = -258

conv\_DEC(83B, 12) = 1199

conv\_BASE(253, 16) = FD

conv\_DEC(11011, 12) = 22477

conv\_BASE(27, 8) = 33

conv\_DEC(11011, 2) = 27

conv\_BASE(512, 8) = 1000

conv\_DEC(1000, 8) = 512

Saludos desde México.

--

**PhD. Felipe de Jesús Martínez Vargas**

**PROFESOR POR HORAS DE FACIMAR Y FCAM**

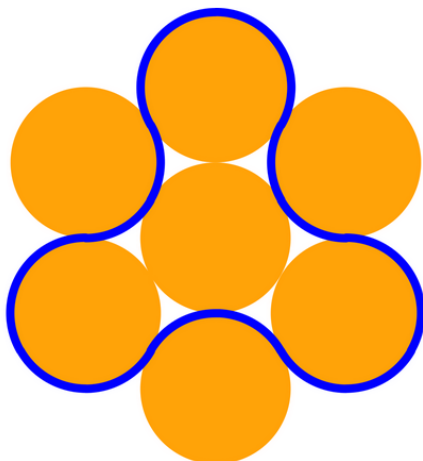
**Cél: 314 10 330 49**

Wolfgang Alvermann sent a mail treating a problem from [1] „Die durchbohrte Kugel” – “The pierced sphere”. I visited the respective website and found “The Rosette”. I couldn’t resist extending the problem (see below) and sent it to Wolfgang ...

## The Rosette

Wolfgang Alvermann & Josef Böhm

<https://www.spektrum.de/raetsel/hemmes-mathematische-raetsel-die-rosettenlinie/1758224>



Die sieben Kreise der Rosette haben alle einen Radius von zehn Zentimetern. Wie lang ist die blaue Linie?

The seven circles of the rosette all have a radius of ten centimeters. What is the length of the blue line?

Das ist nicht allzu schwierig. Aber ich wollte die Figur zeichnen – ohne jeden Kreisbogen einzeln zu definieren.

This is not too difficult. But I wanted to draw the figure - without defining each arc separately.

Zwei VECTOR-Befehle sollten für die Rosette reichen (mit einem gemeinsamen Parameter)

Two VECTOR commands should be sufficient (with one common parameter for the arcs).

$$x^2 + y^2 = r^2$$

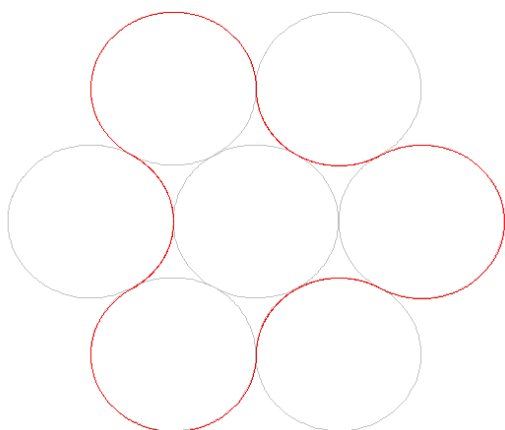
$$\text{VECTOR}\left(\left[\left(x - 2 \cdot r \cdot \cos\left(\frac{k \cdot \pi}{3}\right)\right)^2 + \left(y - 2 \cdot r \cdot \sin\left(\frac{k \cdot \pi}{3}\right)\right)^2 = r^2, k, 0, 5\right]\right)$$

$$\text{VECTOR}\left(\left[\left[2 \cdot r \cdot \cos\left(\frac{\pi \cdot (2 \cdot k - 1)}{3}\right) + r \cdot \cos\left(\frac{\pi \cdot (2 \cdot k + 2 \cdot t - 1)}{3}\right), 2 \cdot r \cdot \sin\left(\frac{\pi \cdot (2 \cdot k - 1)}{3}\right) + r \cdot \sin\left(\frac{\pi \cdot (2 \cdot k + 2 \cdot t - 1)}{3}\right)\right], k, 1, 3\right]\right)$$

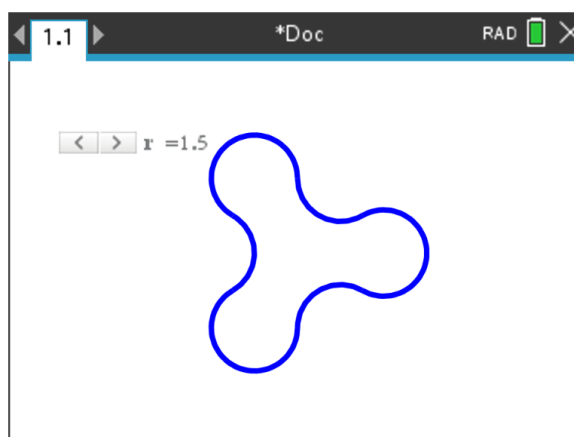
$$\text{VECTOR}\left(\left[\left[2 \cdot r \cdot \cos\left(\frac{\pi}{3} \cdot (2 \cdot k)\right) + r \cdot \cos\left(\frac{2 \cdot \pi \cdot (k + 2 \cdot t)}{3}\right), 2 \cdot r \cdot \sin\left(\frac{\pi}{3} \cdot (2 \cdot k)\right) + r \cdot \sin\left(\frac{2 \cdot \pi \cdot (k + 2 \cdot t)}{3}\right)\right], k, 1, 3\right]\right)$$

Introduce a slider for r in the plot window before plotting.

$$1 \leq t \leq 2$$



DERIVE



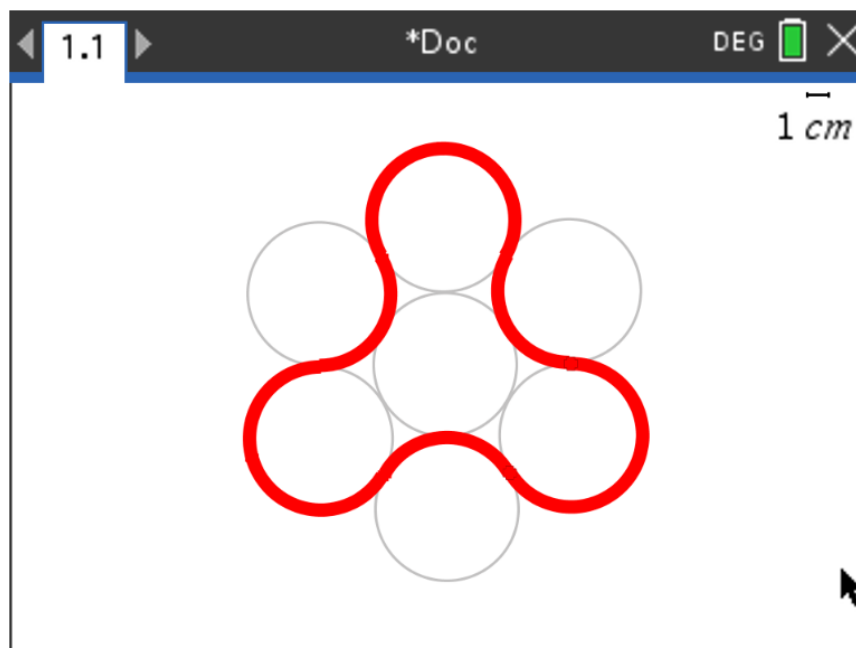
TI-Nspire (including a slider for r)

Mit dem TI-Nspire kann ich diese Folge von Kreisbögen nicht am Stück zeichnen (lassen). So muss ich jeden Bogen einzeln als Parameterkurve definieren.

With TI-Nspire, I cannot draw the curve as sequences of arcs. So, I have to define each one individually as in its parameter form.

Wenn wir es ohne Rechnung machen wollen, dann verwenden wir das Geometrie-Werkzeug.

If we want to do without calculation, then we can use the Geometry tool.



Next day I received Wolfgang's answer. He had been busy with the Rosette earlier. This is what Wolfgang sent. Like me he was not satisfied with finding only the answer to the given question, but added his own questions (including the answers, of course).

- Geben Sie die Koordinaten der Mittelpunkte der äußeren Kreise sowie deren Berührungspunkte an!
- Wie lang ist die blau ausgezogene Länge  $L_R$ ?
- Welche Fläche  $A_R$  umschließt diese Linie?

- Give the coordinates of the centers of the outer circles as well as their osculation points!
- What is the length of the blue curve  $L_R$ ?
- What is the area  $A_R$  enclosed by this curve?

#### Lösung / Solution a)

$$M_1 = (0; 2r) \quad M_2 = (-\sqrt{3}r; r) \quad M_3 = (-\sqrt{3}r; -r)$$

$$M_4 = (0; -2r) \quad M_5 = (\sqrt{3}r; -r) \quad M_6 = (\sqrt{3}r; r)$$

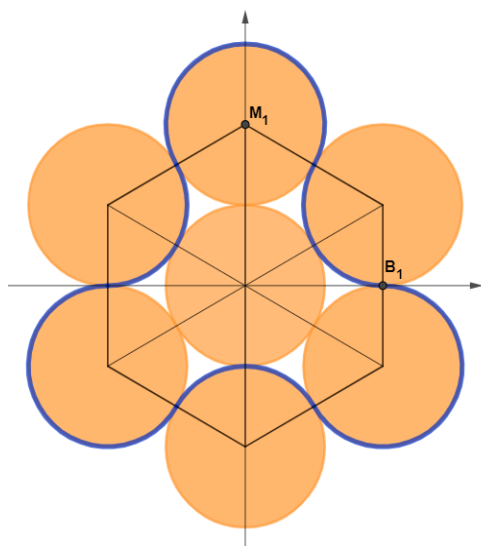
$$B_1 = (\sqrt{3}r; 0) \quad B_2 = \left(\frac{\sqrt{3}}{2}r; \frac{3}{2}r\right) \quad B_3 = \left(-\frac{\sqrt{3}}{2}r; \frac{3}{2}r\right)$$

$$B_4 = (-\sqrt{3}r; 0) \quad B_5 = \left(-\frac{\sqrt{3}}{2}r; -\frac{3}{2}r\right) \quad B_6 = \left(\frac{\sqrt{3}}{2}r; -\frac{3}{2}r\right)$$

**Lösung / Solution b)**

Verbindet man die Mittelpunkte der außen liegenden Kreise, so erhält man ein regelmäßiges Sechseck mit der Seitenlänge  $L = 2r$ . Die Diagonalen unterteilen das Sechseck in sechs gleichseitige Dreiecke mit ebenfalls  $L = 2r$ .

Es ist leicht zu erkennen, dass die Linie genau drei Kreise umschließt  $\rightarrow L_R = 6\pi \cdot r$ .



Connecting the centers of the outer circles gives a regular hexagon with side length  $L = 2r$ . Its diagonals divide the hexagon in six equilateral triangles with sides  $L = 2r$ .

It is easy to see that the curve encloses exactly three circles  $\rightarrow L_R = 6\pi \cdot r$ .

**Lösung / Solution c)**

Wie aus der Grafik zu sehen ist, enthält die geschlossene Linie vier Kreise und sechs von Kreisen eingeschlossene „Dreispitze“

Die Summe der Kreise:  $A_K = 4 \cdot \pi \cdot r^2$

Die „Dreispitze“ erhält man, indem man von einem gleichseitigen Dreieck einen Halbkreis abzieht.

Die Summe der Dreispitze:  $A_{Ds} = 6 \cdot \left( \sqrt{3} \cdot r^2 - \frac{\pi}{2} \cdot r^2 \right) = 3 \cdot r^2 \cdot (2 \cdot \sqrt{3} - \pi)$

Die Addition der Teilflächen ergibt:  $A_R = A_K + A_{Ds}$

$$A_R = (\pi + 6 \cdot \sqrt{3}) \cdot r^2$$

We see that the area consists of four circles and six “tricorn” formed by three circular arches each.

Sum of the circles:  $A_C = 4 \cdot \pi \cdot r^2$

Sum of the tricorns:  $A_{tc} = 6 \cdot \left( \sqrt{3} \cdot r^2 - \frac{\pi}{2} \cdot r^2 \right) = 3 \cdot r^2 \cdot (2 \cdot \sqrt{3} - \pi)$

Total sum:  $A_R = A_C + A_{tc}$

$$A_R = (\pi + 6 \cdot \sqrt{3}) \cdot r^2$$

Links:

[1] <https://www.spektrum.de/raetsel/>

[2] Auch sehr reichhaltig (in German): <https://www.zahlenjagd.at/>

(von dort stammt die Rosette ursprünglich / There you can find the original Rosette problem.)



We had already some contributions on Cryptography in earlier DNLs: Vigenère (DNL39), RSA (DNL42 and DNL70), Merkle-Hellman (DNL42), Hill encryption (DNL103). In Klaus Schmech's book „*Nicht zu knacken*“ I found the *Double Die Encryption*. This encryption method was much used in the intelligence service of the former German Democratic Republic. It is one of the easiest methods which can be carried out with paper and pencil. My Internet research resulted in a lot of interesting resources (see below). So, I decided to make a try programming this method for *Derive* and *TI-Nspire* as well.

## Double Die Encryption

Josef Böhm, Würmla, Austria

The encryption procedure is easy to explain: We intend to encrypt the following message

**“This is a sample text in order to demonstrate the procedure”**

We need two keywords (with different lengths and relatively prime numbers of letters).

I choose “computer” and “programmers”. (Schmech<sup>[1]</sup> recommends lengths between 20 and 25 letters and the lengths shouldn't have common divisors. Moreover, they shouldn't be divisors of the length of message to be encoded.)

First of all, I remove the spaces and write the complete text either in upper case or in lower case letters. (This is not necessary but I believe that this makes it more difficult to break the encrypted message – especially in German.)

The message reads now:

“thisisasampletextinordertodemonstratetheprocedure”

Then I arrange the message in matrix form as shown below. The number of columns is as many as the number of letters of the first keyword. (One should take care to choose keyword and message such that some cells in the last row of the matrix remain empty.)

The first encryption step is to arrange the letters of the keyword in alphabetic order and then rearrange the columns of the matrix in this new order – giving the right matrix below.

| c | o | m | p | u | t | e | r |
|---|---|---|---|---|---|---|---|
| 1 | 4 | 3 | 5 | 8 | 7 | 2 | 6 |
| t | h | i | s | i | s | a | s |
| a | m | p | l | e | t | e | x |
| t | i | n | o | r | d | e | r |
| t | o | d | e | m | o | n | s |
| t | r | a | t | e | t | h | e |
| p | r | o | c | e | d | u | r |
| e |   |   |   |   |   |   |   |

| c | e | m | o | p | r | t | u |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| t | a | i | h | s | s | s | i |
| a | e | p | m | l | x | t | e |
| t | e | n | i | o | r | d | r |
| t | n | d | o | e | s | o | m |
| t | h | a | r | t | e | t | e |
| p | u | o | r | c | r | d | e |
| e |   |   |   |   |   |   |   |

We read the text column by column and receive the first encryption:

[t, a, t, t, t, p, e, a, e, e, n, h, u, , i, p, n, d, a, o, , h, m,  
i, o, r, r, , s, l, o, e, t, c, , s, x, r, s, e, r, , s, t, d, o,  
t, d, , i, e, r, m, e, e, ]

The letters appended with the spaces removed result in the following intermediate message (first die):

“tatttpeaeeenhui pndaohmiorrsloetcsxrserstdotdiermee“

The same procedure as before is applied to it using the second keyword:

| p | r | o | g | r | a | m | m | e | r  | s  |
|---|---|---|---|---|---|---|---|---|----|----|
| 7 | 8 | 6 | 3 | 9 | 1 | 4 | 5 | 2 | 10 | 11 |
| t | a | t | t | t | p | e | a | e | e  | n  |
| h | u | i | p | n | d | a | o | h | m  | i  |
| o | r | r | s | l | o | e | t | c | s  | x  |
| r | s | e | r | s | t | d | o | t | d  | i  |
| e | r | m | e | e |   |   |   |   |    |    |

| a | e | g | m | m | o | p | r | r | r  | s  |
|---|---|---|---|---|---|---|---|---|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| p | e | t | e | a | t | t | a | t | e  | n  |
| d | h | p | a | o | i | h | u | n | m  | i  |
| o | c | s | e | t | r | o | r | l | s  | x  |
| t | t | r | d | o | e | r | s | s | d  | i  |
|   |   | e |   |   | m | e | r | e |    |    |

The encoded message reads (column by column again, without spaces):

“pdotehcttpsreeaedaototiremthoreaursrtnlseemsdnixi”.

It can be grouped for easier sending and reading giving finally the completely encoded message:

“pdoteh cttpsr eeaeda ototir emthor eaursr tnsee msdnix i“.

### How to decipher this message?

Let’s assume that we have received the message from above and we would like to decode it.

We must know both keywords (**computer**, **programmers**).

All what we have to do is to reverse the procedure:

| a | e | g | m | m | o | p | r | r | r  | s  |
|---|---|---|---|---|---|---|---|---|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| p | e | t | e | a | t | t | a | t | e  | n  |
| d | h | p | a | o | i | h | u | n | m  | i  |
| o | c | s | e | t | r | o | r | l | s  | x  |
| t | t | r | d | o | e | r | s | s | d  | i  |
|   |   | e |   |   | m | e | r | e |    |    |

| p | r | o | g | r | a | m | m | e | r  | s  |
|---|---|---|---|---|---|---|---|---|----|----|
| 7 | 8 | 6 | 3 | 9 | 1 | 4 | 5 | 2 | 10 | 11 |
|   |   |   | t |   | p |   |   | e |    |    |
|   |   |   | p |   | d |   |   | h |    |    |
|   |   |   | s |   | o |   |   | c |    |    |
|   |   |   | r |   | t |   |   | t |    |    |
|   |   |   | e |   |   |   |   |   |    |    |

The message consists of 49 letters. The 2<sup>nd</sup> keyword “programmers” consists of 11 letters. So, we need a 11-column matrix to write the message row by row in five rows leaving six cells empty – but we don’t know which one in the left matrix remain empty!! In the right matrix we leave the last six cells empty.

The columns in the left matrix are in alphabetic order of the second keyword and we want to rearrange them in the correct order.

The 1<sup>st</sup> column is the “a-column”. The “a-column” in the right matrix is on sixth position. This is a column with an empty cell in the last row. So, we need only the first four letters of the message “pdot” to fill the a-column.

The “e” is the ninth letter in “programmers” – again an empty cell in the last row. The next four letters of the message are “ehct” and they fill column 9. We proceed with the “g-column” which must be placed as 4<sup>th</sup> column in the right matrix (5 letters).

Column by column the intermediate message (“tatt...rme”) is reconstructed. There is one fact more to consider: letters can appear multiple times. When one column in the right matrix is placed in its correct position, I substitute the respective letter of the keyword by a “?”.

After having worked through all letters we end up in the intermediate code which must be treated in the same way to receive finally the decoded message.

The following lines show the procedure produced by inserting some DISPLAY commands in the code of function enc(message, keyword):

```
#57: [demo := thisisademoxt, k1 := three, k2 := four]
#58: enc(demo, k1)
[t, h, i, s, i, s, a, d, e, m, o, t, x, t, ]
[[t, h, i, s, i], [s, a, d, e, m], [o, t, x, t, ]]
[e, e, h, r, t]
[4, thre, [s, e, t]]
[[t, h, i, i], [s, a, d, m], [o, t, x, ]]
[e, h, r, t]
[4, thr, [s, e, t, i, m, ]]
[[t, h, i], [s, a, d], [o, t, x]]
[h, r, t]
[2, tr, [s, e, t, i, m, , h, a, t]]
[[t, i], [s, d], [o, x]]
[r, t]
[2, t, [s, e, t, i, m, , h, a, t, i, d, x]]
[[t], [s], [o]]
[t]
[1, , [s, e, t, i, m, , h, a, t, i, d, x, t, s, o]]
#59: setimhatidxtso
```

The following DERIVE functions are provided:

|  |  |
|--|--|
| <code>ddie_enc(message, key)</code>                  | performs one change of columns           |
| <code>ddie_dec(message, key)</code>                  | performs one decoding step               |
| <code>ddie_encode(message, key1, key2)</code>        | performs complete encoding               |
| <code>ddie_decode(message, key1, key2)</code>        | performs complete decoding               |
| <code>group(message, n)</code>                       | rewrites message in n-letter blocks      |
| <code>ungroup(message)</code>                        | ungroups the grouped message             |
| <code>ddie_encode_gr(message, key1, key2, b1)</code> | returns the grouped encoded message      |
| <code>ddie_decode_gr(message, key1, key2)</code>     | returns the plain text of a grouped code |

I don't want to print the complete DERIVE code. Instead of this here is a short sample session.

```
plain := thisisasampletextinordertodemonstratetheprocedure
[key1 := computer, key2 := programmers]
[DIM(key1), DIM(key2)] = [8, 11]
mess := ddie_encode(plain, key1, key2)
mess := pdotehcttpsreeaedaototirenthoreaursrtnlseemsdnixi
ddie_decode(mess, key1, key2) = thisisasampletextinordertodemonstratetheprocedure
```

As you can see below, the functions are working with longer messages, too. The line breaks don't cause any problems, fortunately.

```
longtxt :=
Weneedttwokeywords(withdifferentlengthsandrelativelyprimenumbersofletters~
).Ichoosecomputerandprogrammers.(Schmehrecommendslengthsbetween20and25le~
ttersandthelengthsshouldnthavecommondivisors.Moreover,theyshouldntbedivi~
sorsofthelengthofmessagetobeencoded.)
```

```
enclong := encode(longtxt, key1, key2)
```

```
enclong :=
```

```
dbsm1heeb.semawgihe,hgdftto1nnokaoihe1ddtefessfeeahvyneutmtlesddritmre0e~
sngIsdctednd.rneuaeontfrgcoheamysirhstir2gs(etnletr1albrsstouewe.Wrmrso~
eosn1tlwseeoenpvorsdhtgebreopdnouepdesnintmee1hoch.eoorMemo)hdtcemr(~
Sormhnsii5tteeentovwcvfctnev2hraaho)
```

```
decode(enclong, key1, key2)
```

```
Weneedttwokeywords(withdifferentlengthsandrelativelyprimenumbersofletters).~
Ichoosecomputerandprogrammers.(Schmehrecommendslengthsbetween20and25lett~
ersandthelengthsshouldnthavecommondivisors.Moreover,theyshouldntbediviso~
rsofthelengthofmessagetobeencoded.)
```

I must admit that programming the deciphering function took me more time than the enciphering one. I worked with transposed matrices because I found treating rows more comfortable than manipulating columns. Grouping of the cipher and ungrouping was no problem.

```
ddie_encode_gr(plain, key1, key2, 6)
```

```
pdoteh cttpsr eeaeda ototir emthor eaursr tn1see msdnix i
```

```
ddie_decode_gr(pdoteh cttpsr eeaeda ototir emthor eaursr tn1see msdnix i, key1, key2)
```

```
thisisasampletextinordertodemonstratetheprocedure
```

It was clear for me that I should program this encryption method for TI-Nspire, too. I had to consider some differences in treating strings and matrices but – as you can see on the following pages – it works properly.

```
Define enc(mess,key)=
```

```
Func
```

```
:Local lm,lk,mat,sk,pos,messn,key_1,i,j
```

```
:lm:=dim(mess):lk:=dim(key)
```

```
:mess:=seq(mid(mess,i,1),i,1,lm)
```

```
:mess:=augment(mess,seq("",i,1,lk-  
mod(lm,lk)))
```

```
:mat:=newMat(floor(((lm)/(lk)))+1,lk)
```

```
:For i,0,floor(((lm)/(lk)))
```

```
: For j,1,lk
```

```
: mat[i+1,j]:=mess[i*lk+j]
```

```
: EndFor
```

```
:EndFor
```

```
:mat:=matT
```

```
:key_1:=seq(ord(mid(key,i,1)),i,1,lk)
```

```
:sk:=sort(key_1)
```

```
:sk:=seq(char(sk[i]),i,1,dim(sk))
```

```
:mess:={}
```

```
:For i,1,lk
```

```
: pos:=inString(key,sk[i])
```

```
: key:=left(key,pos-1)&"?"&right(key,lk-pos)
```

```
: For j,1,floor(((lm)/(lk)))+1
```

```
: mess:=augment(mess,{mat[pos,j]})
```

```
:EndFor::EndFor
```

```
:messn:=mess[1]
```

```
:For i,2,dim(mess)
```

```
: If mess[i]≠"":messn:=messn&mess[i]
```

```
:EndFor
```

```
:messn
```

```
:EndFunc
```

Applying twice function enc:

```
Define encode(mess,key1,key2)=
```

```
Func
```

```
:mess:=enc(mess,key1):enc(mess,key2)
```

```
:EndFunc
```

```
Define dec(mess,key)=
```

```
Func
```

```
:Local lm,lk,mat,sk,pos,messn,nc,nf,key_1,i,j
```

```
:lm:=dim(mess):lk:=dim(key)
```

```
:© number of columns & of full cells in last col
```

```
:nc:=ceiling(((lm)/(lk))):nf:=mod(lm,lk)
```

```
:key_1:=seq(ord(mid(key,i,1)),i,1,lk)
```

```
:sk:=sort(key_1)
```

```
:sk:=seq(char(sk[i]),i,1,dim(sk))
```

```
:messn:=seq("",i,1,lk)
```

```
:For i,1,lk
```

```
: pos:=inString(key,sk[i])
```

```
: If pos>nf Then
```

```
: messn[pos]:=left(mess,nc-1)
```

```
: Else
```

```
: messn[pos]:=left(mess,nc)
```

```
: EndIf
```

```
: If nf=0:messn[pos]:=left(mess,nc)
```

```
: mess:=right(mess,dim(mess)-dim(messn[pos]))
```

```
: key:=left(key,pos-1)&"?"&right(key,lk-pos)
```

```
: If pos>nf:messn[pos]:=messn[pos]&" "
```

```
:EndFor
```

```
:mess:=""
```

```
:For i,1,nc
```

```
: For j,1,lk
```

```
: If mid(messn[j],i,1)≠" "
```

```
: mess:=mess&mid(messn[j],i,1)
```

```
:EndFor:EndFor:EndFunc
```

Applying twice function dec:

```
Define decode(code,key1,key2)=
```

```
Func
```

```
:code:=dec(code,key2):dec(code,key1)
```

```
EndFunc
```

The functions for grouping and ungrouping the encoded message followed by the combination of coding and decoding grouped messages.

|   |  |
|---|--|
| <pre> Define group(txt,n)= Func :Local txtbl :txtbl="" :While dim(txt)&gt;n : txtbl:=txtbl&amp;left(txt,n)&amp;" " : txt:=right(txt,dim(txt)-n) EndWhile :txtbl&amp;txt :EndFunc </pre> | <pre> Define ungroup(txt)= Func :Local txtbl,i :txtbl="" :For i,1,dim(txt) : If mid(txt,i,1)≠" ":txtbl:=txtbl&amp;mid(txt,i,1) :EndFor :EndFunc </pre> |
|---|--|

Define encode\_gr(mess\_key1\_key2\_n)=group(encode(mess\_key1\_key2\_n),n)

Define decode\_gr(mess\_key1\_key2)=decode(ungroup(mess\_key1\_key2))

I'd like to close the contribution showing some sample screen shots of TI-Nspire

```

key1:="programmers":key2:="computer"                                "computer"
message:="thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
        "thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
coded:=encode(mess,key1,key2)
        "sas.asiiteildtathreadstenoittpnhnto easilsvwerhremotlhleswcarwieldtocidyweurrxd.wrepteeimp"
decode(coded,key1,key2)
        "thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
coded2:=encode(mess,key2,key1)
        "phaooxteehy.cetetenms.waseuorerdrtlwrtsomtephlaisntescivdiaailehdaetndilssiwnirenderptrwliw"
decode(coded2,key2,key1)
        "thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
coded:=encode_gr(message,key1,key2,6)
        "sas.asiiteildtathreadstenoittpnhnto easilsvwerhremotlhleswcarwieldtocidyweurrxd.wrepteeimp"
decode_gr(coded,key1,key2)
        "thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
|

```

```

numbs:="123456789012345123456789012345"           "123456789012345123456789012345"
k1:="seven":k2:="eleven"                             "eleven"
{dim(numbs),dim(k1),dim(k2)}                         {30,5,6}
encode(numbs,k1,k2)                                   "245132451379068790682451324513"
decode("245132451379068790682451324513",k1,k2)     "123456789012345123456789012345"
encode(message,k1,k2)
    "hst.myeowlolaewclrdahemeastrithdeenepo.ixdheowaeisrdaetnilerwdsriatlsrwrirstuntcieetppnstv"
decode("hst.myeowlolaewclrdahemeastrithdeenepo.ixdheowaeisrdaetnilerwdsriatlsrwrirstuntcieetppnstv",k
    "thisisasampletextinordertodemonstratetheprocedure.wewilltrywithderiveandwithnspirecasaswell."
encode(numbs,key1,key2)                               "670153934202312988614545375421"
decode("670153934202312988614545375421",key1,key2)  "123456789012345123456789012345"
encode_gr(numbs,key1,key2,5)                          "67015 39342 02312 98861 45453 75421"
decode_gr("67015 39342 02312 98861 45453 75421",key1,key2)  "123456789012345123456789012345"
encode_gr(numbs,k2,k1,8)                              "79182046 85970241 53135264 352431"
decode_gr("79182046 85970241 53135264 352431",k2,k1)  "123456789012345123456789012345"

```

Inserted Disp commands let us follow the procedure (here for one step decoding):

```

dec("setimhatidxtso","three")
    {3,4}
    {"e","e","h","r","t"}
    4
    imhatidxtso
    thr?e
    {"","","","set",""}
    5
    hatidxtso
    thr??
    {"","","","set","im "}
    2
    idxtso
    t?r??
    {"","hat","","set","im "}
    3
    tso
    t????
    {"","hat","idx","set","im "}

```

```

Invalid data type
Define dec(mess,key)=
Func
Local lm,lk,mat,sk,pos,messn,nc,nf,key_1,i,j
lm:=dim(mess):lk:=dim(key)
© number of columns & number of full cells in last
col
nc:=ceiling(lm/lk):nf:=mod(lm,lk)
Disp {nc,nf}
key_1:=seq(ord(mid(key,i,1)),i,1,lk)
sk:=sort(key_1)
sk:=seq(char(sk[i]),i,1,dim(sk))
Disp sk
messn:=seq("",i,1,lk)
For i,1,lk
    pos:=inString(key,sk[i])
    If pos>nf Then
        messn[pos]:=left(mess,nc-1)
    Else
        messn[pos]:=left(mess,nc)
    EndIf
Disp pos

```

I found many internet resources treating this encryption method (next page).

[1] Klaus Schmeh, Nicht zu knacken, Hanser 2012

<https://infsec.de/doppelwuerfel/>

<https://kryptografie.de/kryptografie/chiffre/doppelter-spaltentausch.htm>

<http://informatik.rostfrank.de/info/lex09/doppelwuerfel.html#ref7>

<https://www.geogebra.org/m/gwm7fbzu>

<https://www.dcode.fr/two-square-cipher> (many languages)

<http://www.easysurf.cc/cipher3.htm>

<https://www.tutorialspoint.com/cryptography/index.htm>

<https://www.pdf-tutorial.com/security/79-course-introduction-to-cryptography>

[https://www.tutorialspoint.com/cryptography\\_with\\_python/cryptography\\_with\\_python\\_tutorial.pdf](https://www.tutorialspoint.com/cryptography_with_python/cryptography_with_python_tutorial.pdf)

### Mail from Tunesia:

Dear colleague,

Following the publication of my paper entitled: "A new vision of the universe evolution" some readers have made the following remark : the author of the Qur'an could be an exceptionally intelligent person, referring to the prophet of Islam. But did anyone who lived in an era (6-7th century AD) and in a place (Arabia) characterized by obscurantism and ignorance imagine that at the end of the solar system , the sun becomes like a swelling ball ("**When the sun becomes like a swelling ball**" *Qur'an 81: 1*) and there will be a disintegration of mountains and a boiling of the seas and oceans because of the increase of the temperature on the globe ("**I Swear)... By the Canopy Raised High, And by the boiling sea.**" *Qur'an 52: 5-6*). *Without comment!*

The paper that I have broadcasted, is in fact a part of an extended document entitled : "STUDY OF SCIENTIFIC ASPECTS IN THE QUR'AN"

[https://www.researchgate.net/publication/332082410\\_STUDY\\_OF\\_SCIENTIFIC\\_ASPECTS\\_IN\\_THE\\_QUR'AN](https://www.researchgate.net/publication/332082410_STUDY_OF_SCIENTIFIC_ASPECTS_IN_THE_QUR'AN)

it consists of four parts :

- the first part concerns the context of the Quran's revelation.
- the second part concerns the verses relating to general matters such as the organization of communities in the animal world, the process of fecundation, the optimisation in creation, etc... .
- the third part deals with verses relating to life science. We study, in particular, the formation of milk in the animal organism, the origin of living beings and Darwin's theory of evolution.
- the fourth part concerns the verses focussing on Astronomy (To have a quick idea of this part I designed a powerpoint presentation of its contents.

[https://www.researchgate.net/publication/331962994\\_The\\_Future\\_of\\_Solar\\_System](https://www.researchgate.net/publication/331962994_The_Future_of_Solar_System)

- A powerpoint presentation of my last paper entitled " THE EXPERIENCE ON THE BRINK OF DEATH AND THE HEREAFTER " will be posted online in the coming months. This paper is also available on the researchgate website:

[https://www.researchgate.net/publication/326960357\\_THE\\_EXPERIENCE\\_ON\\_THE\\_BRINK\\_OF\\_DEATH\\_AND\\_THE\\_HEREAFTER\\_THE\\_VERDICT](https://www.researchgate.net/publication/326960357_THE_EXPERIENCE_ON_THE_BRINK_OF_DEATH_AND_THE_HEREAFTER_THE_VERDICT)

Kamel Ben Salem

Professor of Data analysis at the Department of Computer Science at the Faculty of Science,  
Tunis, 2092 El Manar II - Tunisia;

[http://www.fst.rnu.tn/kbs/All\\_fichiers/Home.html](http://www.fst.rnu.tn/kbs/All_fichiers/Home.html)



### Résumé 1 – Part 3

## Résolution symbolique, résolution numérique d'équations et de systèmes d'équations Rappels et compléments en équations différentielles

### 2- Méthode du point fixe et méthode de Newton

Commençons par une équation scalaire du type  $g(x) = x$  afin d'introduire le principe de contraction de Banach.

**2.1 Théorème** Soit  $g$  une fonction continue sur l'intervalle fermé  $I$ , telle que  $g(x) \in I \quad \forall x \in I$ .

a) Alors  $g$  admet (au moins) un point fixe  $r$ , i.e.  $\exists r \in I$  tel que  $g(r) = r$ .

b) Supposons en plus  $g$  dérivable sur l'intérieur de  $I$  et qu'il existe une constante  $K$  telle que  $|g'(x)| \leq K < 1 \quad \forall x \in I$ . Alors l'équation  $g(x) = x$  admet une unique solution dans  $I$ . De plus, la suite des approximations successives définie par  $x_{n+1} = g(x_n)$ ,  $n = 0, 1, 2, \dots$  converge vers cette solution, quel que soit le point de départ  $x_0 \in I$ .

(La raison pour laquelle on parle de « contraction », c'est que, par le théorème de la valeur moyenne, on a, si  $x$  et  $y \in I$ , qu'il existe un nombre  $z$  entre  $x$  et  $y$  tel que  $g(x) - g(y) = g'(z)(x - y)$ , d'où,  $|g(x) - g(y)| \leq K|x - y|$ . Comme  $K$  est inférieur à 1, cela signifie que  $g$  « rapproche » les points, donc  $g$  est une « contraction »).

Preuve : a) en esquissant un graphe, on « prouve » la partie a) puisque si  $I = [a, b]$  et si  $g(a) = a$  et/ou si  $g(b) = b$ , on a déjà (au moins) un point fixe. Puisque l'image de  $I$  est contenue dans  $I$  par hypothèse, alors si  $g(a) > a$  et si  $g(b) < b$ , alors la continuité de  $g$  (ou le théorème de la valeur intermédiaire appliqué à la fonction  $h(x) = x - g(x)$ ) donne le résultat.

b) Pour l'unicité, notons que si l'on a 2 points fixes, disons  $x$  et  $y$ , alors le théorème de la valeur moyenne implique l'existence d'un  $z$  entre  $x$  et  $y$  tel que  $g(x) - g(y) = g'(z)(x - y)$ . Ainsi

$$|x - y| = |g(x) - g(y)| = |g'(z)(x - y)| = |g'(z)||x - y| \leq K|x - y|$$

d'où  $|x - y|(1 - K) \leq 0$ . Mais  $1 - K > 0$ , de sorte que  $x = y$ . Passons à l'existence. Notez que si l'on définit une suite par  $x_{n+1} = g(x_n)$ ,  $n = 0, 1, 2, \dots$  où  $x_0 \in I$  est quelconque, alors cette suite est bien définie puisque l'image de  $g$  reste dans  $I$ . De plus, si cette suite converge vers un nombre  $r$  disons, alors ce nombre est forcément un point fixe de  $g$ . En effet, par 2 théorèmes d'analyse, la limite d'une suite convergente d'éléments d'un intervalle fermé appartient à cet intervalle et une fonction continue « transporte » les suites convergentes, donc

$$r = \lim_{n \rightarrow \infty} x_{n+1} = \lim_{n \rightarrow \infty} g(x_n) = g\left(\lim_{n \rightarrow \infty} x_n\right) = g(r).$$

Il reste donc à démontrer que la suite, définie par  $x_{n+1} = g(x_n)$ ,  $n = 0, 1, 2, \dots$ , converge. On note que

$x_n = x_0 + (x_1 - x_0) + (x_2 - x_1) + \dots - x_{n-1}$ ). Par conséquent, la suite  $\{x_n\}_{n=0}^{\infty}$  convergera si et seu-

lement si la série  $\sum_{j=0}^{\infty} (x_{j+1} - x_j)$  converge. Pour démontrer cela, c'est l'hypothèse de contraction qu'on

applique et la « bonne vieille » série géométrique. En effet :

$$\begin{aligned}
 |x_2 - x_1| &= |g(x_1) - g(x_0)| \leq K|x_1 - x_0| \\
 |x_3 - x_2| &= |g(x_2) - g(x_1)| \leq K|x_2 - x_1| \leq K^2|x_1 - x_0| \\
 &\vdots \\
 &\vdots \\
 |x_{j+1} - x_j| &\leq K^j|x_1 - x_0|
 \end{aligned}$$

Puisque  $K < 1$ , la série géométrique  $|x_1 - x_0| \sum_{j=0}^{\infty} K^j$  converge et le critère de comparaison implique la convergence de la série  $\sum_{j=0}^{\infty} |x_{j+1} - x_j|$ . Et la convergence absolue impliquant la convergence, la preuve est terminée. ♦

**2.2 Exemples** Plusieurs détails seront donnés en classe pour les exemples qui suivent.

**2.2.1** L'équation  $x = \frac{x}{2} + \frac{1}{x}$  est facile à résoudre puisqu'équivalente à  $x^2 = 2$ . Avec  $g(x) = \frac{x}{2} + \frac{1}{x}$  et  $x_0 = 1$ , ça convergera vers  $1.4142136 \approx \sqrt{2}$ .

**2.2.2** L'équation  $x^3 - 3x + 1 = 0$  possède 3 racines réelles et distinctes (nous avons montré comment les trouver en mode exact en 1.6), dont l'une est située entre 1 et 2. On verra que  $g(x) = \frac{x^3 + 1}{3}$  ou  $g(x) = \frac{1}{3 - x^2}$  ne fonctionnent pas mais que  $g(x) = \frac{3}{x} - \frac{1}{x^2}$  fonctionne (autrement dit que cette fonction satisfait les hypothèses du théorème 2.1b) et on trouve 1.53209).

**2.2.3** L'équation  $x = 2 \cos x$  possède une solution entre 0 et  $\pi/2$ . Mais  $g(x) = 2 \cos x$  ne fonctionne pas tandis que  $g(x) = \frac{x + 2 \cos x}{2}$  fonctionne (on trouve 1.02987). Nous illustrons ici avec *Derive* et sa fonction `FIXED_POINT(f, x, x0, n)` qui itère  $f(x)$ , partant de  $x_0$ ,  $n$  fois, donnant, une fois simplifié ou mieux, approximé, un vecteur à  $n + 1$  composantes.

#1: `FIXED_POINT(2 * COS(x), x, 1.5, 10)`

#2: `[1.5, 0.1414744033, 1.980018354, -0.7957914253, 1.399439154, 0.3410396031, 1.884814923, -0.6177664155, 1.630348433, -0.1190338253, 1.985847670]`

#3: `FIXED_POINT`  $\left( \frac{x + 2 \cdot \text{COS}(x)}{2}, x, 1.5, 10 \right)$

#4: `[1.5, 0.8207372016, 1.092050620, 1.006691570, 1.038005274, 1.026942146, 1.030909002, 1.029493846, 1.029999627, 1.029818978, 1.029883515]`

Le lecteur intéressé verra à utiliser le système de son choix et/ou à programmer lui-même ces différentes méthodes. Utilisée entre 1999 et 2011 à l'ÉTS, les calculatrices symboliques TI-89 Titanium ou Voyage 200 de la compagnie Texas Instruments montraient déjà (figure 2.1) comment on s'en sort très bien quand on peut mettre en mémoire des fonctions et les rappeler facilement :

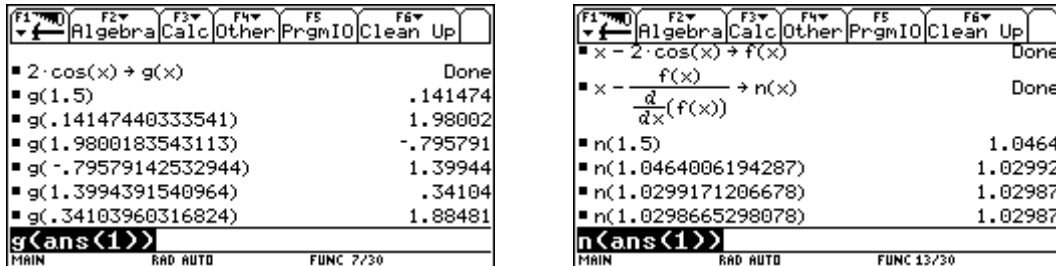


Figure 2.1

On peut aussi visualiser le fait que  $g(x) = 2 \cos x$  ne fonctionne pas : ici avec Nspire CX CAS.

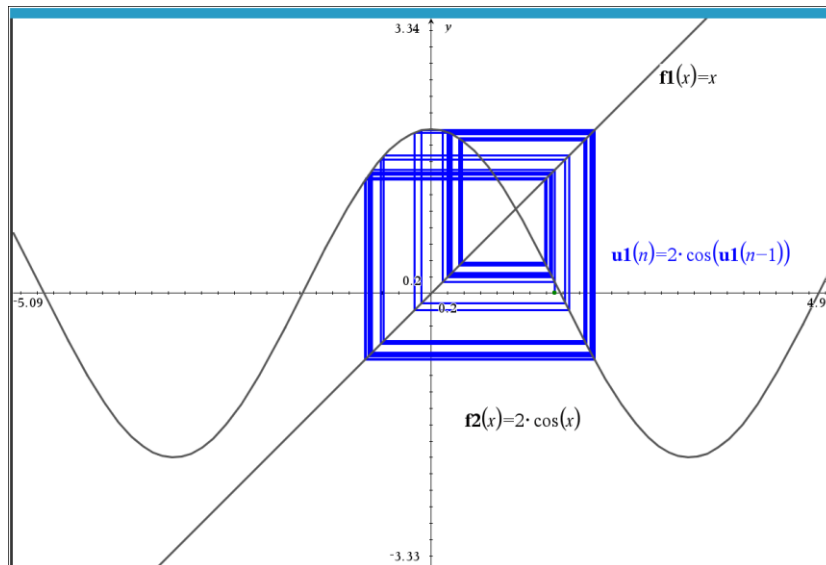


Figure 2.2

Les fonctions « fixed\_point » et « newton » sont incluses dans la librairie kit\_ets\_mb :

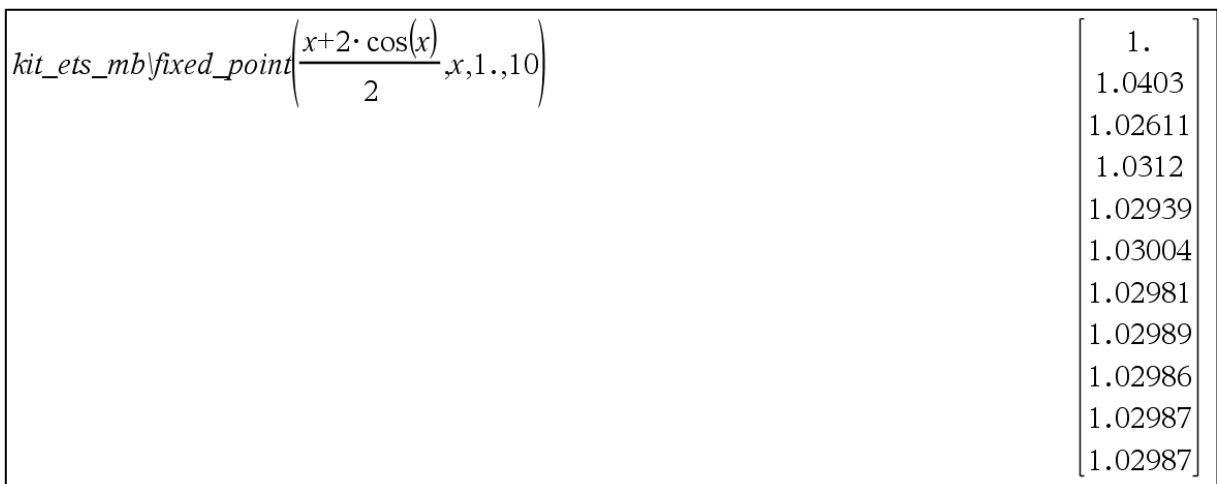


Figure 2.3

**2.3 Remarque et rappel** Rappelons la formule de Taylor pour les fonctions d'une variable : si  $f$  est une fonction dérivable jusqu'à l'ordre  $n + 1$  dans un intervalle ouvert  $I$  contenant le point  $a$ , alors pour chaque  $x \in I$ , on a

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x-a)^n + R_n(x)$$

où le reste (l'erreur)  $R_n(x)$  est calculé par  $R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1}$  pour un  $c$  entre  $x$  et  $a$ .

En effet, en posant

$$g(t) = f(x) - f(t) - f'(t)(x-t) - \frac{f''(t)}{2!}(x-t)^2 - \dots - \frac{f^{(n)}(t)}{n!}(x-t)^n - R_n(x) \frac{(x-t)^{n+1}}{(x-a)^{n+1}}$$

on vérifie aisément que  $g$  satisfait le théorème de Rolle ( $g(a) = g(x) = 0$ ). Donc, il existe un nombre  $c$  entre  $a$  et  $x$  tel que  $g'(c) = 0$ . Mais  $g'(t) = -\frac{f^{(n+1)}(t)}{n!}(x-t)^n + R_n(x)(n+1)\frac{(x-t)^n}{(x-a)^{n+1}}$ , et il suffit de poser  $g'(c) = 0$  et de résoudre.

**2.4 Théorème (Méthode de Newton et vitesse de convergence)** Soit  $f$  une fonction admettant une dérivée seconde continue sur l'intervalle  $I$  et possédant un zéro  $r$  à l'intérieur de  $I$ , donc  $f(r) = 0$ . Supposons qu'il existe des nombres positifs  $m$  et  $M$  tels que  $|f'(x)| \geq m$  et  $|f''(x)| \leq M$  sur  $I$ .

Si  $x_1 \in I$  est suffisamment proche de  $r$ , disons  $|x_1 - r| < \frac{2m}{M}$ , si

$$(1) \quad x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 1, 2, \dots$$

alors on a

$$|x_{n+1} - r| < \frac{M}{2m}(x_n - r)^2 \quad \text{et} \quad \lim_{n \rightarrow \infty} x_n = r.$$

Preuve : la formule de Taylor implique qu'il existe un nombre  $c$  entre  $x_n$  et  $r$  tel que

$$f(r) = f(x_n) + f'(x_n)(r - x_n) + \frac{f''(c)}{2!}(r - x_n)^2$$

Puisque  $f(r) = 0$ , on obtient  $|x_{n+1} - r| = \left| x_n - \frac{f(x_n)}{f'(x_n)} - r \right| = \left| \frac{f''(c)}{2f'(x_n)} \right| (x_n - r)^2 \leq \frac{M}{2m}(x_n - r)^2$ .

Montrons, par récurrence sur  $n$ , qu'on a  $|x_n - r| \leq \frac{2m}{M} \left( \frac{M}{2m} |x_1 - r| \right)^{2^{n-1}}$ . Si  $n = 1$ , c'est vrai puisque

$|x_1 - r| \leq \frac{2m}{M} \left( \frac{M}{2m} |x_1 - r| \right)^{2^0} = |x_1 - r|$ . Supposons que c'est vrai pour  $n = k$ , donc supposons que

$|x_k - r| \leq \frac{2m}{M} \left( \frac{M}{2m} |x_1 - r| \right)^{2^{k-1}}$ . Montrons que c'est vrai pour  $n = k + 1$ . On a

$$|x_{k+1} - r| \leq \frac{M}{2m} (x_k - r)^2 \leq \frac{M}{2m} \left( \frac{2m}{M} \left( \frac{M}{2m} |x_1 - r| \right)^{2^{k-1}} \right)^2 = \frac{2m}{M} \left( \frac{M}{2m} |x_1 - r| \right)^{2^k} \text{ et la récurrence est terminée.}$$

Ainsi, en choisissant  $|x_1 - r| < \frac{2m}{M}$ , on a  $\lim_{n \rightarrow \infty} |x_n - r| = 0$ , ce qui termine la preuve. ♦

**2.5 Exemples** (détails en classe) : puisque la méthode de Newton consiste à itérer la fonction

$$x - \frac{f(x)}{f'(x)},$$

on retrouve encore cette méthode programmée dans *Derive*. L'exemple 2.2.3 de tantôt donne ceci lorsqu'on approxime la fonction  $\text{NEWTON}(f, x, x0, n)$  où  $f(x) = 0$ .

```
#5:  NEWTON(x - 2·COS(x), x, 1.5, 10)
#6:  [1.5, 1.046400619, 1.029917120, 1.029866529, 1.029866529, 1.029866529,
      1.029866529, 1.029866529, 1.029866529, 1.029866529]
```

**Figure 2.4**

Et nous avons « exporté » cette fonction vers Nspire :

|   |  |
|---|--|
| <code>kit_ets_mb\newton(x-2·cos(x),x,1.5,10)</code> | 1.5<br>1.0464<br>1.02992<br>1.02987<br>1.02987<br>1.02987<br>1.02987<br>1.02987<br>1.02987<br>1.02987<br>1.02987 |
|---|--|

**Figure 2.5**

Les méthodes du point fixe et de Newton s'étendent aux systèmes d'équations et des logiciels de calcul possèdent déjà des fonctions programmées pour les exécuter (évidemment, les preuves de ces résultats font intervenir des normes de matrices jacobiniennes). Ainsi, si l'on cherche à résoudre le système d'équations (système non linéaire et très souvent non polynomial) donné par

$$\begin{cases} f_1(x_1, x_2, \dots) \\ f_2(x_1, x_2, \dots) \\ \vdots \\ f_n(x_1, x_2, \dots) \end{cases}$$

et si l'on veut appliquer la méthode de Newton, on devra itérer le vecteur  $\mathbf{x} - \mathbf{J}^{-1}\mathbf{F}$  où

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ \dots]$$

$$\mathbf{J} = \left[ \frac{\partial f_i}{\partial x_j} \right]_{i,j=1}^n$$

en partant d'un point de départ  $\mathbf{x}_0$ . Ici la matrice  $\mathbf{J}$  est la matrice jacobienne du système. Évidemment, une implémentation va remplacer le calcul de  $\mathbf{J}^{-1}\mathbf{F}$  par une réduction de Gauss-Jordan de la matrice augmentée  $[\mathbf{J}]$  : dont on extraira la dernière colonne en ayant eu soin d'insérer le point avant la réduction! Nous allons illustrer en classe avec les systèmes suivants :

$$2.5.1 \quad \begin{cases} x^2 + y + z - 3 = 0, & x + y^2 + z - 3 = 0, & x + y + z^2 - 3 = 0 \end{cases}$$

Notons que ce système est résoluble analytiquement et qu'il possède huit solutions réelles dont deux faciles à trouver par inspection. Il est particulièrement intéressant de faire ici un lien avec le calcul à plusieurs variables. Les deux cylindres paraboliques  $z = 3 - x^2 - y$ ,  $z = 3 - x - y^2$  se rencontrent le long d'une courbe qui consiste en deux paraboles et on vérifiera que  $r_1(t) = [t, t, 3 - t^2 - t]$ ,  $r_2(t) = [t, -t + 1, 2 - t^2 + t]$  en sont des paramétrisations. En portant dans l'équation de la troisième surface, on obtient l'équation  $t^4 + 2t^3 - 5t^2 - 4t + 6 = 0$  dont les solutions sont  $\pm\sqrt{2}$ , 1 et -3. Cela procure les huit solutions! Voici une illustration des deux paraboles provenant de l'intersection des deux premiers cylindres :

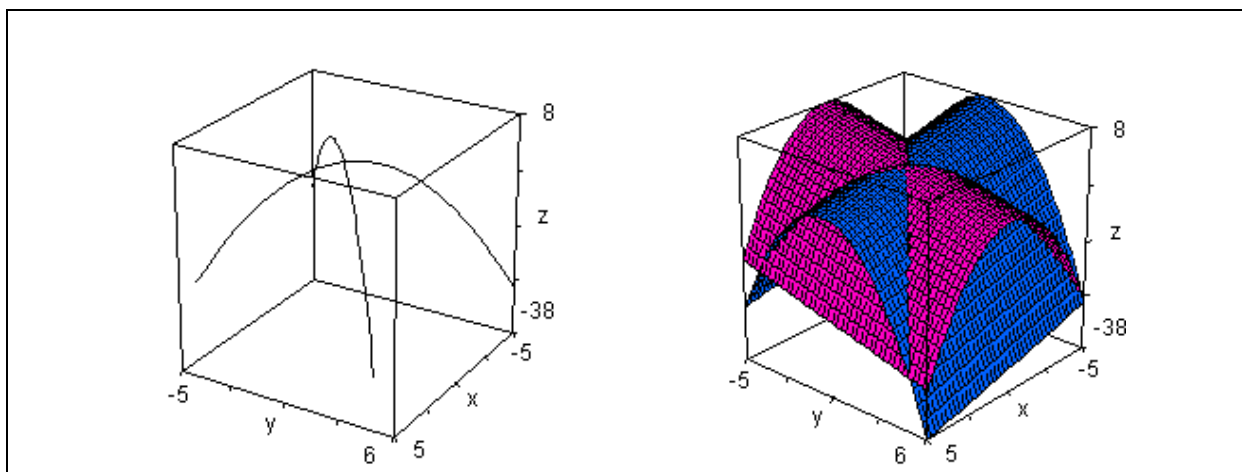


Figure 2.6

Voici les deux paraboles « frappant » le troisième cylindre, à savoir  $y = 3 - z^2 - x$  (chacune des 2 paraboles rencontre le cylindre 4 fois) :

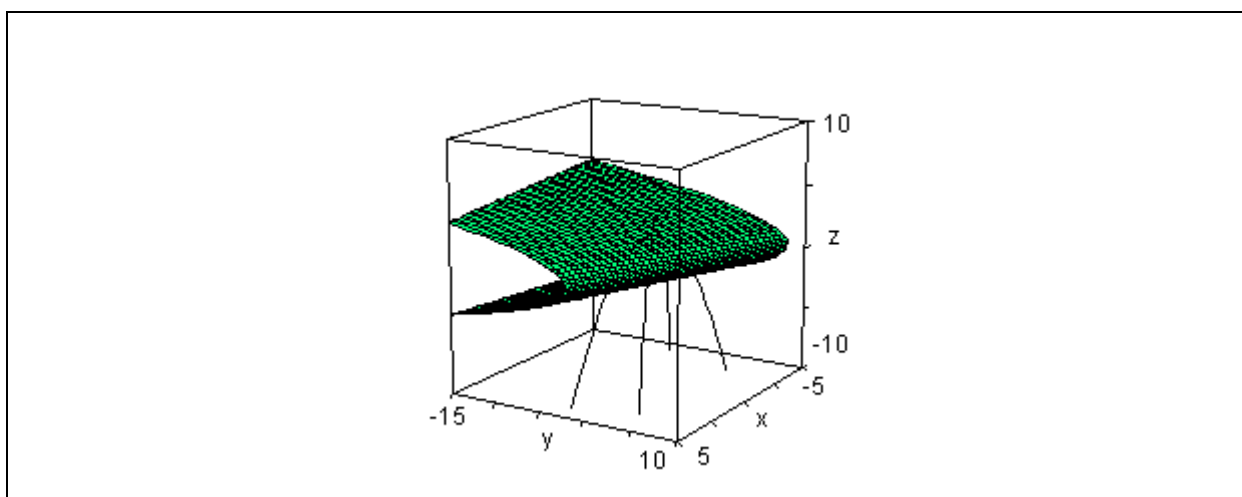


Figure 2.7

Quant au fait que ce système polynomial possède bien huit solutions, cela est démontrable mathématiquement : les logiciels de calcul symbolique sont munis de fonctions genre « base de Gröbner » qui réduisent un système polynomial en un autre équivalent, mais dont la particularité est de contenir une équation à une seule inconnue. On peut donc y appliquer le théorème fondamental de l'algèbre et, ensuite, substituer dans les autres équations. Cela permet de savoir le nombre exact de solutions. Dans l'exemple qui nous intéresse, regardons ce que *Derive* 6 donne. Sa fonction « solutions » donne les 8 solutions sous forme matricielle :

$$\begin{aligned} \#1: \quad \text{syst} &:= \left[ x^2 + y + z - 3, x + y^2 + z - 3, x + y + z^2 - 3 \right] \\ \#2: \quad \text{SOLUTIONS}(\text{syst}, [x, y, z]) &= \begin{bmatrix} 1 & 1 & 1 \\ -3 & -3 & -3 \\ \sqrt{2} & \sqrt{2} & 1 - \sqrt{2} \\ \sqrt{2} & 1 - \sqrt{2} & \sqrt{2} \\ \sqrt{2} + 1 & -\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & \sqrt{2} + 1 & -\sqrt{2} \\ -\sqrt{2} & -\sqrt{2} & \sqrt{2} + 1 \\ 1 - \sqrt{2} & \sqrt{2} & \sqrt{2} \end{bmatrix} \end{aligned}$$

Figure 2.8

Et sa fonction « groebner\_basis » permet de voir qu'il y a exactement six valeurs différentes de  $z$  (mais en substituant chacune d'elles, huit valeurs sont générées pour  $x$  et  $y$ ) :

$$\begin{aligned} \#3: \quad & \text{GROEBNER\_BASIS}(\text{syst}, [x, y, z]) \\ \#4: \quad & \left[ z^6 - 10 \cdot z^4 + 4 \cdot z^3 + 19 \cdot z^2 - 8 \cdot z - 6, y \cdot (2 \cdot z^2 - 4) + z^4 - 5 \cdot z^2 + 6, \right. \\ & \left. y^2 - y - z^2 + z, x + y + z^2 - 3 \right] \\ \#5: \quad & \text{SOLUTIONS}(z^6 - 10 \cdot z^4 + 4 \cdot z^3 + 19 \cdot z^2 - 8 \cdot z - 6, z) \\ \#6: \quad & [1, -3, \sqrt{2}, \sqrt{2} + 1, -\sqrt{2}, 1 - \sqrt{2}] \end{aligned}$$

Figure 2.9

$$2.5.2 \quad \begin{cases} x \cos 2 + 2 \sin y = \frac{1}{2} \\ x y = 2 \end{cases}$$

Le système précédent est réductible à une seule équation à une inconnue mais il doit être nécessairement résolu numériquement et possède une infinité de solutions comme le lecteur pourra le vérifier.

2.5.3 Voici une séance de *Derive* pour le système

$$\begin{cases} 3xy - 2x^2 + 4\sin(y) + 6 = 0 \\ 3x^2 - 2xy^2 + 3\cos(x) + 4 = 0 \end{cases}$$

Ici, la fonction exécutant la méthode de Newton requiert une notation vectorielle : dans *Derive*, on écrit et approxime  $\text{NEWTONS}(\mathbf{f}, \mathbf{x}, \mathbf{x0}, n)$  où  $\mathbf{f}$  est le vecteur des expressions à annuler,  $\mathbf{x}$  est le vecteur des variables,  $\mathbf{x0}$  est le vecteur du point de départ et  $n$  le nombre entier d'itérations. Pour avoir une idée de point de départ, nous traçons, dans une même fenêtre, chacune des 2 courbes définies par les équations ci-haut : la courbe #8 est celle en rouge. Cela est prévisible analytiquement puisqu'il est impossible que  $x = 0$  dans cette équation...!

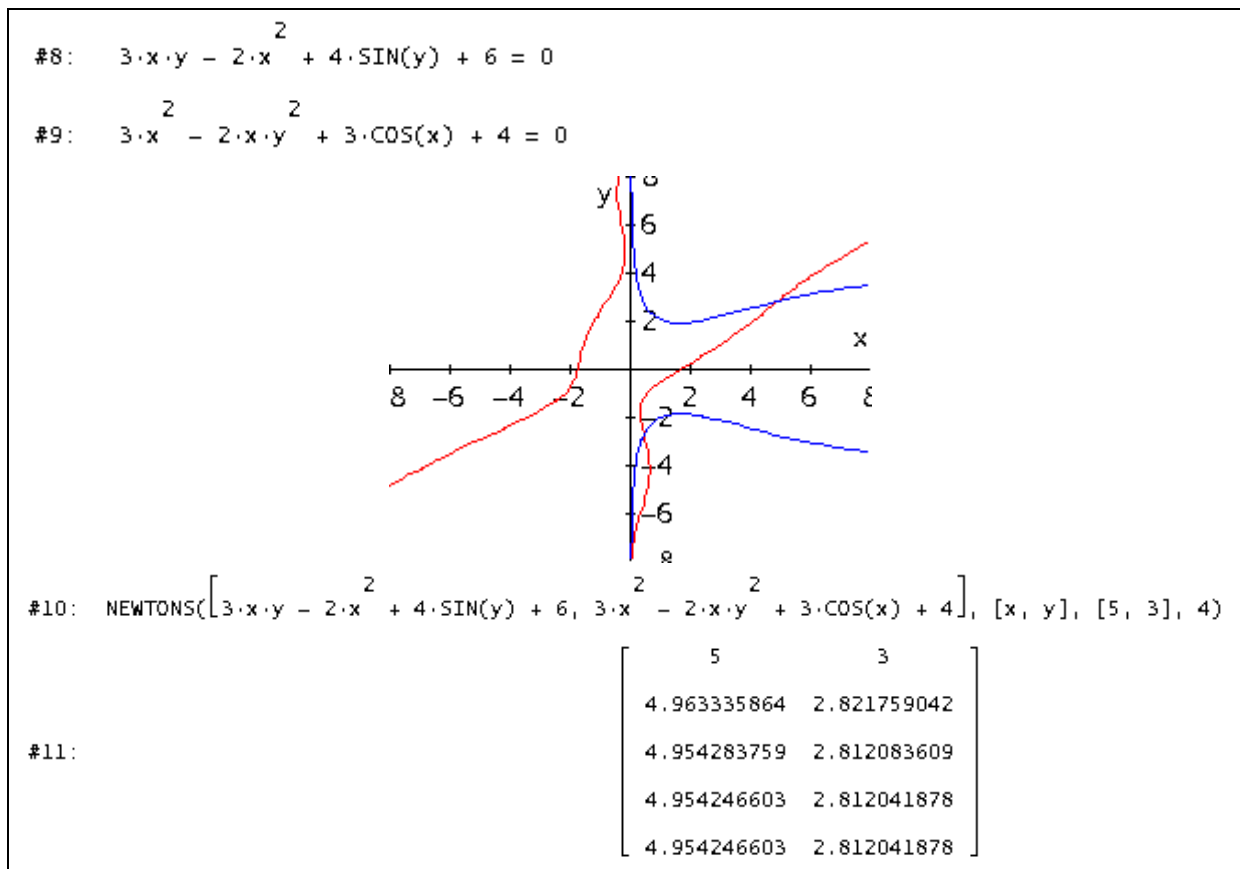


Figure 2.10

On peut aussi faire cet exemple avec Nspire et produire encore le graphe des courbes puisque du graphisme implicite 2D n'est pas vraiment requis.

En effet, il est possible de résoudre pour  $x$  la première équation, ce qui donne deux branches et faire de même pour  $y$  dans la seconde équation, donnant aussi deux branches : la figure 2.11 montre le travail et l'utilisation de la fonction  $\text{Newtons2}$  de la bibliothèque `kit_est_mb`. Notez que le solveur de Nspire, sans point de départ, trouve la solution située dans le quatrième quadrant. Et en lui donnant un point de départ « adéquat », le solveur trouve bien la solution située dans le premier quadrant.



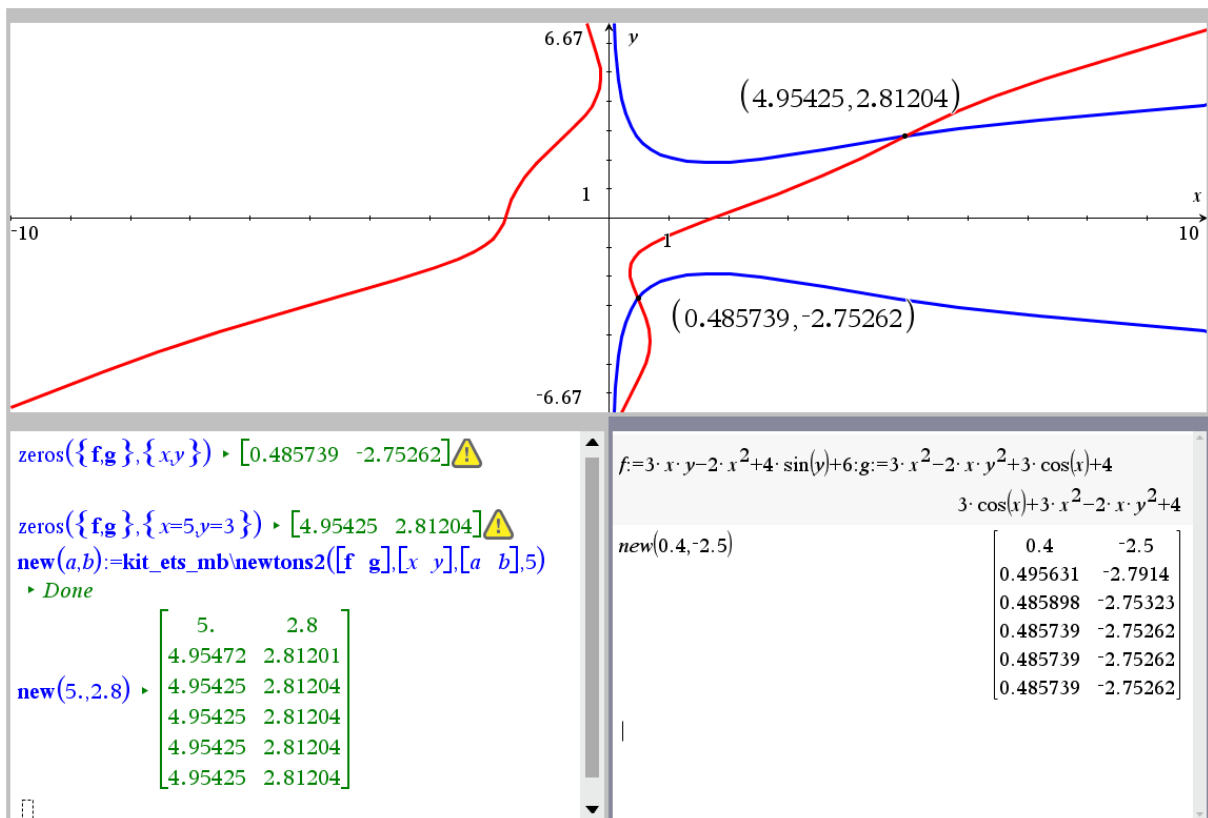


Figure 2.11

Will be continued

Maybe you will remember Wolfgang Pröpper's contribution on Lottery results from DNL 121.

This email refers to it:

Lieber Josef,

vielleicht hast Du es schon gesehen. Gestern gab es wieder einen sehr seltenen Lotto-Wurf, nämlich einen Zwilling (9, 10) gepaart mit einem Drilling (44, 45, 46). Bei der Gelegenheit ist mir aber eingefallen, dass ich mich noch gar nicht zum letzten newsletter gemeldet und mich vor allem bedankt habe. Auch wenn mein Beitrag kurz ist, hattest Du doch einige Mühe und Arbeit damit. Also nochmals, vielen Dank.

Einen schönen Gruß an Euch Beide,  
Wolfgang

Dear Josef,

maybe you have already seen it. Yesterday there was again a very rare lotto result, namely a twin (9, 10) paired with a triplet (44, 45, 46). On this occasion, however, I remembered that I have not yet reported to the last newsletter and thanked you above all. Even though my contribution is short, you had some effort and work with it. Once again, many thanks.

Best regards to you both

Wolfgang

## A Survey over Distributions – well known and not so well known 3

Bhuvanesh Bhatt & Josef Böhm & Felipe de Jesús Martínez Vargas

(All TI-Nspire functions are contained in the library mathtool.tns.)

### General Discrete Distribution

This is the distribution of a discrete number of outcomes of a random variable with different probabilities.

Example:

|          |     |     |     |     |     |                 |
|----------|-----|-----|-----|-----|-----|-----------------|
| $X_i$    | 1   | 3   | 5   | 7   | 9   |                 |
| $p(X_i)$ | 0.2 | 0.2 | 0.4 | 0.1 | 0.1 | $\sum p(X_i)=1$ |

The density function in *DERIVE*:

```
pdf_gendiscr(rv, pv, x, d, i) :=
  Prog
  d := DIM(rv)
  If d ≠ DIM(pv)
    RETURN "Wrong dimensions"
  If Σ(pv) ≠ 1
    RETURN "Check probabilities!"
  i := 1
  Loop
  If i > d
    RETURN 0
  If x = rv↓i
    RETURN pv↓i
  i :=+ 1
```

`pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.2], 1.5) = Check probabilities!`

`pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 1.5) = 0`

`pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 3) = 0.2`

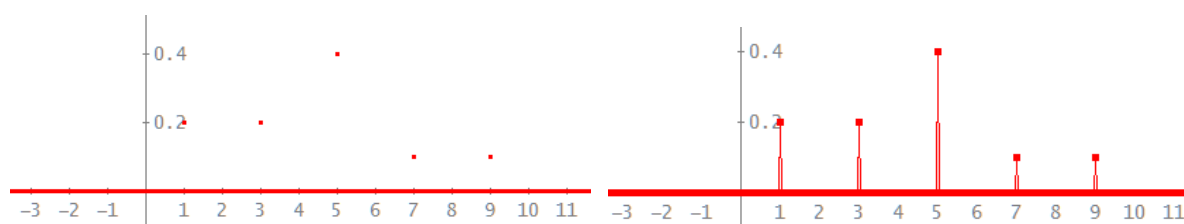
`pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 5) = 0.4`

`pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], x)`

`TABLE(pdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], x), x, -4, 12, 0.05)`

Plotting the density function `pdf_gendiscr([1,3,5,7,9], [0.2,0.2,0.4,0.1,0.1],`

`x)` results in plotting the  $x$ -axis only.



It can be plotted in form of a table (points not connected and connected).

The cumulative density function:

```

cdf_gendiscr(rv, pv, x, d, i, p) :=
  Prog
  d := DIM(rv)
  If d ≠ DIM(pv)
    RETURN "Wrong dimensions"
  If Σ(pv) ≠ 1
    RETURN "Check probabilities!"
  If x < rv↓1
    RETURN 0
  [p := 0, i := 1]
  Loop
  If x < rv↓i
    RETURN p
  [p := p + pv↓i, i := i + 1]
  If i > d exit
  RETURN 1

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 2) = 0.2

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 0.9) = 0

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 3) = 0.4

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 5.4) = 0.8

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 7) = 0.9

```

```

cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], 9) = 1

```

```

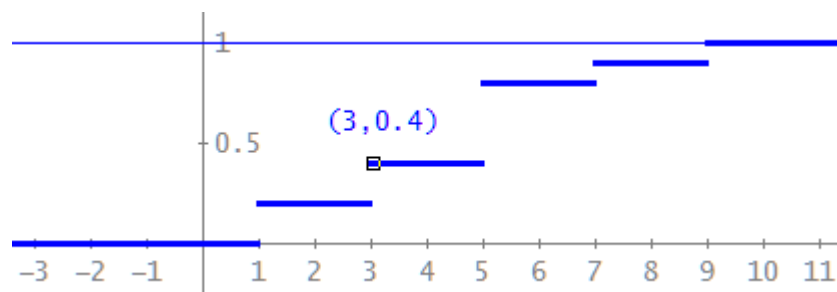
cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], x)

```

```

TABLE(cdf_gendiscr([1, 3, 5, 7, 9], [0.2, 0.2, 0.4, 0.1, 0.1], x), x, -4, 12, 0.05)

```



The function works properly but its graph results in the horizontal line  $y = 1$ . The graph of the table shows the steps and the cursor “jumps” correctly.

I tried a second version using *DERIVE*'s indicator function  $\text{CHI}(a,x,b)$  which gave a correct graph of the function – but with hidden secrets!!

```

cdf_gendiscr2(rv, pv, x, d, i, p) :=
  Prog
  d := DIM(rv)
  If d ≠ DIM(pv)
    RETURN "Wrong dimensions"
  If Σ(pv) ≠ 1
    RETURN "Check probabilities!"
  If x < rv↓1
    RETURN 0
  RETURN Σ(Σ(pv↓j, j, 1, i)·χ(rv↓i, x, rv↓(i + 1), 1), i, 1, d - 1) + χ(rv↓d, x, ∞, 1, 1)

```



## Discrete Uniform Distribution

Wikipedia: The **discrete uniform distribution** is a distribution wherein a finite number of values in  $[a, b]$  are equally likely to be observed; every one of  $n$  values has equal probability  $1/n$ .

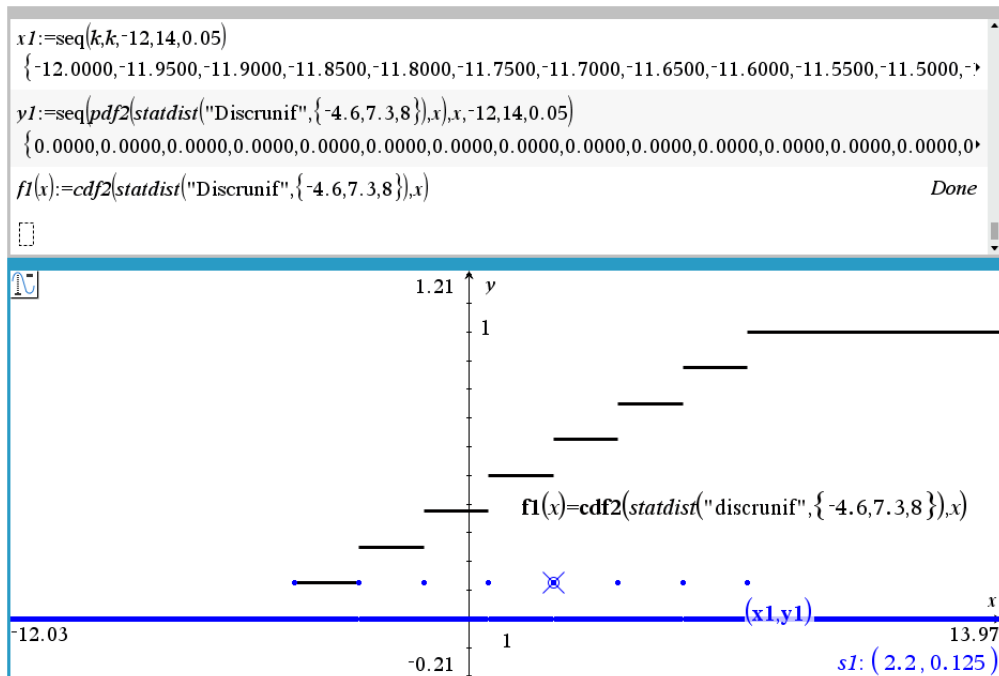
Density function: 
$$f(X) = \begin{cases} \frac{1}{n} & \text{if } X = x_i = a + \frac{b-a}{n-1}(i-1) \text{ for } i=1, \dots, n \\ 0 & \text{else} \end{cases}$$

Distribution function: 
$$F(x) = \begin{cases} 0 & x < a \\ \frac{\left\lfloor \frac{(x-a)(n-1)}{b-a} \right\rfloor + 1}{n} & \text{or } \frac{\sum k \mid x_k \leq x}{n} & a \leq x \leq b \\ 1 & x > b \end{cases}$$

|   |               |  |   |
|---|---------------|--|---|
| $pdf2(statdist("Discrunif", \{1, 6, 6\}), 4)$             | $\frac{1}{6}$ | <pre>pdf2 48/77 ENDIF If disttype="discrunif" Then aa:=pars[1] bb:=pars[2] nn:=pars[3] For ii,1,nn If xx=aa+ (bb-aa)/(nn-1) * (ii-1):Return 1/nn EndFor Return 0 ENDIF</pre> |   |
| $pdf2(statdist("Discrunif", \{1, 6, 6\}), 3.5)$           | 0             |  |   |
| $pdf2(statdist("Discrunif", \{5, 10, 4\}), \frac{20}{3})$ | $\frac{1}{4}$ |  |   |
| $pdf2(statdist("Discrunif", \{-4, 6, 7, 3, 8\}), 0.5)$    | $\frac{1}{8}$ |  |   |
| $cdf2(statdist("Discrunif", \{1, 6, 6\}), 4.5)$           | $\frac{2}{3}$ |  | <pre>cdf2 46/75 ENDIF If disttype="discrunif" Then aa:=pars[1]:bb:=pars[2]:nn:=pars[3] Return 0, xx &lt; aa Return floor((xx-aa) * (nn-1) / (bb-aa)) + 1 / nn, aa &lt;= xx &lt;= bb Return 1, ENDIF If disttype="triangle" Then aa:=pars[1]:mm:=pars[2]:bb:=pars[3]</pre> |
| $cdf2(statdist("Discrunif", \{5, 10, 4\}), 9)$            | $\frac{3}{4}$ |  |   |
| $cdf2(statdist("Discrunif", \{-4, 6, 7, 3, 8\}), 7)$      | $\frac{7}{8}$ |  |   |
| $cdf2(statdist("Discrunif", \{1, 6, 6\}), 6.5)$           | 1             |  |   |
| $cdf2(statdist("Discrunif", \{1, 6, 6\}), 1)$             | $\frac{1}{6}$ |  |   |
| $cdf2(statdist("Discrunif", \{1, 6, 6\}), 6)$             | 1             |  |   |

$pdf2(statdist("Discrunif", \{5, 10, 4\}), 20/3) = 1/4$  says that there are 4 equally distributed values  $X$  with  $5 \leq X \leq 10$  and we want to know  $p(X = 20/3)$ .

Plotting the pdf needs little work: we define the  $x$ - and  $y$ -values as sequences and plot them. On the screen one can observe the gaps in the blue line on the  $x$ -axis. Tracing the graph gives a jump at each nonzero position. Here:  $p(X = 2.2) = 1/8$ .



DERIVE:

```

pdf_discrunif(a, b, n, x, xx, ii) :=
  Prog
  xx := a
  ii := 1
  Loop
#1:   If xx = x
      RETURN 1/n
      xx := xx + (b - a)/(n - 1)
      ii := + 1
      If ii > n exit
  RETURN 0

```

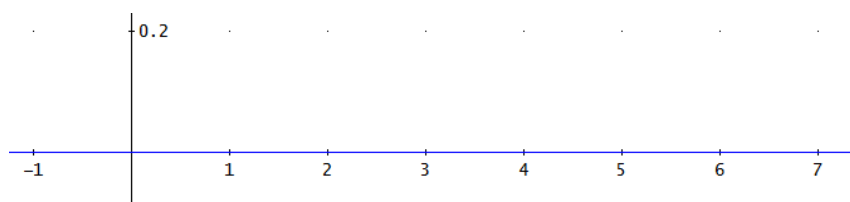
$$\#2: \text{pdf\_discrunif}(1, 6, 6, 2) = \frac{1}{6}$$

$$\#3: \text{pdf\_discrunif}(1, 6, 6, 6) = \frac{1}{6}$$

$$\#4: \text{pdf\_discrunif}(1, 6, 6, 2.5) = 0$$

$$\#5: \text{pdf\_discrunif}(1, 6, 6, 20) = 0$$

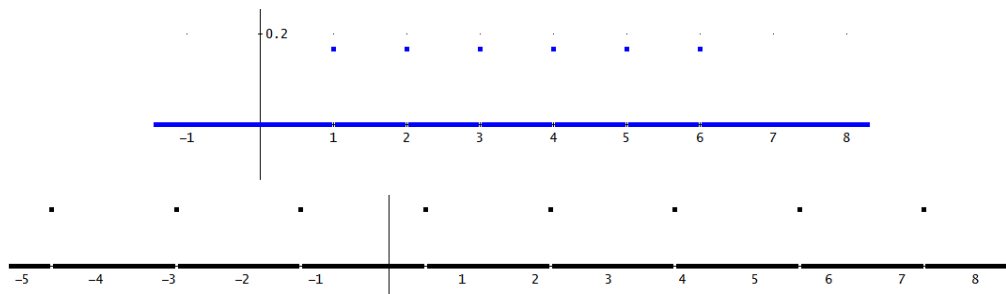
$$\#6: \text{pdf\_discrunif}(1, 6, 6, x)$$



For plotting the pdf it is necessary to produce the TABLE.

```
#7: TABLE(pdf_discrunif(1, 6, 6, x), x, -4, 10, 0.05)
```

```
#8: TABLE(pdf_discrunif(-4.6, 7.3, 8, x), x, -6, 10, 0.05)
```



```
cdf_discrunif(a, b, n, x) :=
```

```
  If x < a
```

```
    0
```

```
#9:   If x ≤ b
```

```
      (FLOOR((x - a)·(n - 1)/(b - a)) + 1)/n
```

```
    1
```

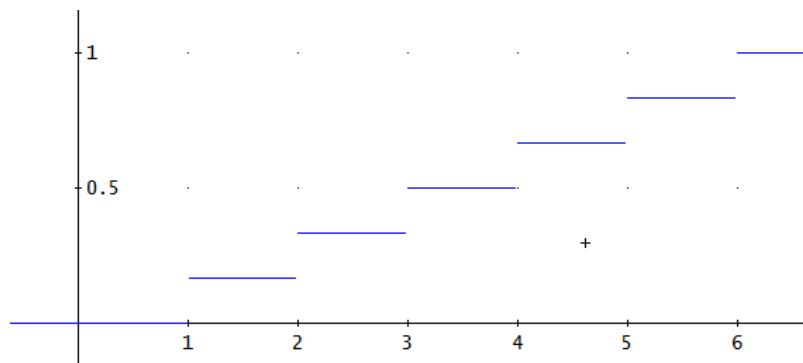
$$\#10: \text{cdf\_discrunif}(1, 6, 6, 4.5) = \frac{2}{3}$$

$$\#11: \text{cdf\_discrunif}(-4.6, 7.3, 8, 7) = \frac{7}{8}$$

$$\#12: \text{cdf\_discrunif}(5, 10, 4, 9) = \frac{3}{4}$$

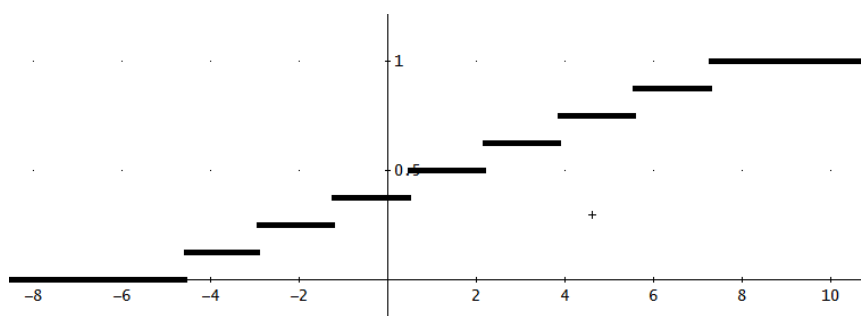
$$\#13: \text{cdf\_discrunif}(1, 6, 6, x)$$

The cdf can be plotted directly, however, the TABLE provides a nicer plot.



$$\#14: \text{cdf\_discrunif}(-4.6, 7.3, 8, x)$$

$$\#15: \text{TABLE}(\text{cdf\_discrunif}(-4.6, 7.3, 8, x), x, -10, 12, 0.05)$$

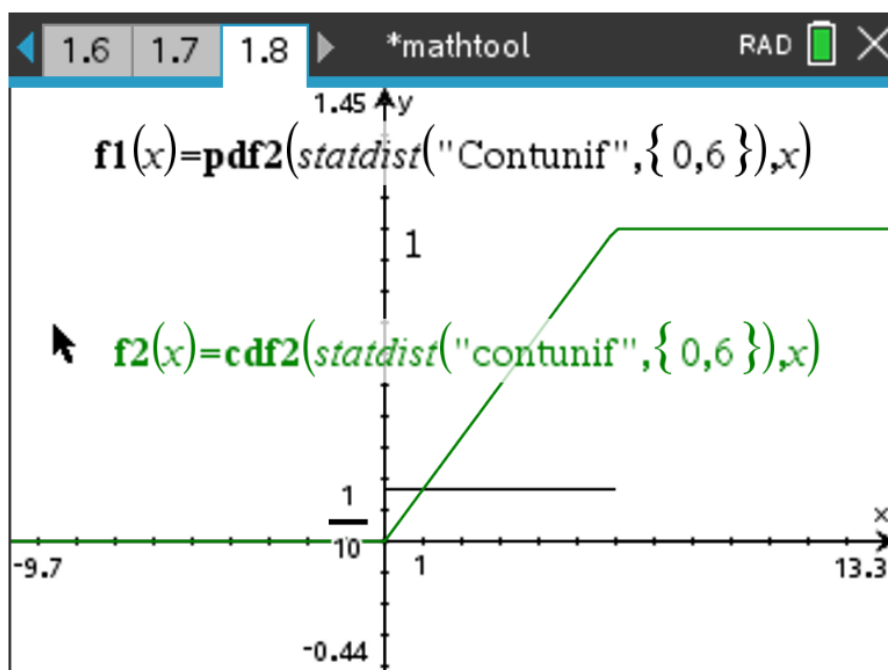


## Continuous Uniform Distribution

Wikipedia: The **continuous uniform distribution** or **rectangular distribution** describes the probability of a random variable  $X$  with an arbitrary outcome that lies between certain bounds  $a$  and  $b$ .

Pdf and cdf can easily be read off from the TI-Nspire functions' code.

|  |                 |   |
|--|-----------------|---|
| $cdf2(statdist("Discrunif",\{-4,6,7,3,8\}),7)$ | $\frac{7}{8}$   | <pre>"pdf2" stored successfully If disttype="contunif" Then aa:=pars[1]:bb:=pars[2] Return { 0, xx&lt;aa 1, aa&lt;=xx&lt;=bb 0, } EndIf If disttype="discrunif" Then aa:=pars[1] bb:=pars[2] nm:=pars[3] For ii,1,nm bb-aa / \ 1 cdf2 EndIf If disttype="contunif" Then aa:=pars[1]:bb:=pars[2] Return { 0, xx&lt;aa xx-aa, aa&lt;=xx&lt;=bb bb-aa, } 1, } EndIf If disttype="discrunif" Then aa:=pars[1]:bb:=pars[2]:nm:=pars[3] Return { 0, xx&lt;aa floor((xx-aa):(nm-1)), } 1, } Return</pre> |
| $cdf2(statdist("Discrunif",\{1,6,6\}),6.5)$    | 1               |   |
| $cdf2(statdist("Discrunif",\{1,6,6\}),1)$      | $\frac{1}{6}$   |   |
| $cdf2(statdist("Discrunif",\{1,6,6\}),6)$      | 1               |   |
| $pdf2(statdist("Contunif",\{0,6\}),3.5)$       | $\frac{1}{6}$   |   |
| $cdf2(statdist("Contunif",\{0,6\}),3.5)$       | $\frac{7}{12}$  |   |
| $pdf2(statdist("Contunif",\{1,6\}),3.5)$       | $\frac{1}{5}$   |   |
| $cdf2(statdist("Contunif",\{1,6\}),3.5)$       | $\frac{1}{2}$   |   |
| $pdf2(statdist("Contunif",\{0,23\}),10)$       | $\frac{1}{23}$  |   |
| $cdf2(statdist("Contunif",\{0,23\}),10)$       | $\frac{10}{23}$ |   |
| $f1(x):=pdf2(statdist("Contunif",\{0,6\}),x)$  | Done            |   |
| $f2(x):=cdf2(statdist("Contunif",\{0,6\}),x)$  | Done            |   |





## StudentT Distribution

**Student's  $t$ -distribution** (or simply the  **$t$ -distribution**) is any member of a family of continuous probability distributions that arise when estimating the mean of a normally-distributed population in situations where the sample size is small and the population's standard deviation is unknown. For sample sizes larger than 20 (e.g. more degrees of freedom), the distribution is almost exactly like the normal distribution. It was developed by English statistician William Sealy Gosset under the pseudonym "Student". (Wikipedia)

Density function: 
$$f_n(t) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi} \Gamma\left(\frac{n}{2}\right)} \left(1 + \frac{t^2}{n}\right)^{-\frac{n+1}{2}}$$

Distribution function: 
$$F_n(t) = I\left(\frac{t + \sqrt{t^2 + n}}{2\sqrt{t^2 + n}}, \frac{n}{2}, \frac{n}{2}\right)$$
 with  $I(z, a, b) = \frac{1}{B(a, b)} \int_0^z t^{a-1} (1-t)^{b-1} dt$

$n$  ... degrees of freedom,  $B(a, b)$  = Beta function (find more about this function on page 45)

The StudentT-pdf is part of Bhuvanesh's mathtool library but the respective cdf is not. I added the cdf for V200 and TI-92 first and proceeded programming the functions for TI-Nspire and DERIVE as well.

I start with *wxMaxima*:

```
(%i1) load(distrib)$
(%i12) float(pdf_student_t(2,5));
(%o12) 0.06509031032621648
(%i13) float(pdf_student_t(-1,10));
(%o13) 0.2303619892291387
(%i14) float(pdf_student_t(.5,.66));
(%o14) 0.2224618592641987
(%i15) float(pdf_student_t(1.6449,10));
(%o15) 0.1042507183470315
(%i16) float(cdf_student_t(2,5));
(%o16) 0.9490302605850708
(%i17) float(cdf_student_t(-1,10));
WARNING: redefining MAXIMA::SIMP-UNIT-STEP in DEFUN
WARNING: redefining MAXIMA::SIMP-POCHHAMMER in DEFUN
(%o17) 0.1704465661510299
(%i20) float(cdf_student_t(.5,.66));
(%o20) 0.6325285614859188
(%i21) float(cdf_student_t(1.6449,10));
(%o21) 0.9344920404057686
```

Example (Felipe de Jesus Martinez Vargas)

Calculate, for a Student's t distribution, the probability that the variable takes a value of 1.6449 to the right of that point. Take as degrees of freedom  $n = 10$ .

Answer: 0.934492 (see above and below):

StudentT for TI-NspireCAS:

The screenshot shows the TI-NspireCAS interface with the StudentT function definition and its application to various parameters. The function definition is as follows:

$$\text{pdf}(\text{statdist}(\text{"Studentt"}, \{3\}, x)) = \frac{6 \cdot \sqrt{3}}{\pi \cdot (x^2 + 3)^2}$$

$$\text{cdf}(\text{statdist}(\text{"Studentt"}, \{3\}, x) | x > 0) = \frac{\sin^{-1}\left(\frac{x}{\sqrt{x^2 + 3}}\right)}{\pi} + \frac{\sqrt{3} \cdot x}{\pi \cdot (x^2 + 3)} + \frac{1}{2}$$

The function is applied to various parameters:

- $\text{pdf}(\text{statdist}(\text{"Studentt"}, \{5\}, 2)) = 0.065090310326$
- $\text{cdf}(\text{statdist}(\text{"Studentt"}, \{5\}, 2)) = 0.949030260585$
- $\text{pdf}(\text{statdist}(\text{"Studentt"}, \{10\}, -1)) = 0.230361989229$
- $\text{cdf}(\text{statdist}(\text{"Studentt"}, \{10\}, -1)) = 0.170446566154$
- $\text{pdf}(\text{statdist}(\text{"Studentt"}, \{0.66\}, 0.5)) = 0.222461856762$
- $\text{cdf}(\text{statdist}(\text{"Studentt"}, \{10\}, 1.6449)) = 0.934492040406$

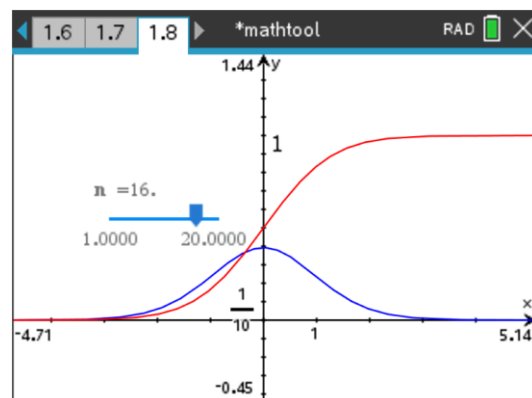
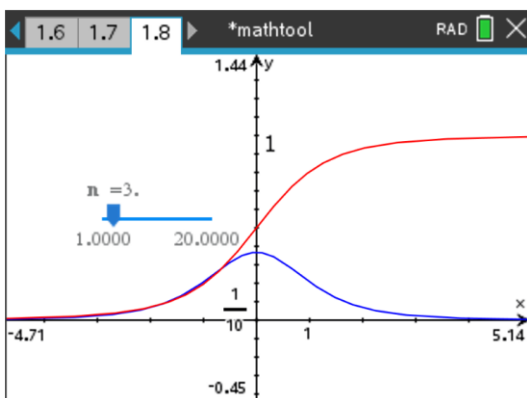
The function definition also includes a conditional statement for the Poisson distribution:

```

If disttype="poisson" Then
  Return  $\frac{\left(\frac{\text{pars}[1]}{\text{pars}[1] + xx^2}\right)^2 \cdot \text{mathtool}\gamma\left(\frac{\text{pars}[1]}{2}\right)}{\sqrt{\text{pars}[1]} \cdot \sqrt{\pi} \cdot \text{mathtool}\gamma\left(\frac{\text{pars}[1]}{2}\right)}$ 
EndIf

```

As the parameter (number in  $\{\}$ ) is the number of degrees of freedom (which is sample size  $- 1$ ) the value 0.66 given in one example does not make much sense – nevertheless, the function works.



I was so fascinated by Bhuvanesh's library and fixed to transfer and accomplish it for TI-Nspire that I did not remember that powerful statistics tools are implemented (see below). But, never mind, it was exciting to produce "self-made" functions.

I will close the "TI-Nspire section" of the StudentT-treatment with another example (from *Cranshaw & Chambers, A Concise Course in A-Level Statistics*):

Ten packets of a particular brand of biscuits are chosen at random and their masses noted. The results (in grams) are 397.3, 399.6, 401.0, 392.9, 396.8, 400.0, 397.6, 392.1, 400.8, 400.6. Assuming that the sample is taken from a normal distributed population with mean mass  $\mu$ , calculate (a) the 95% confidence interval for  $\mu$ , (b) the 99% confidence interval for  $\mu$ .

|   |   |
|---|---|
| tPdf(2,5)   | 0.0651  |
| tCdf(-∞,2,5)  | 0.9490  |
| tPdf(x,3)   | tPdf(x,3)   |
| tPdf(3,3)   | 0.0230  |
| tPdf(-1,10)   | 0.2304  |
| tPdf(0.5, $\frac{2}{3}$ )   | 0.2233  |
| tCdf(-∞, -1,10)   | 0.1704  |
| masses:= $\{397.3,399.6,401.,392.9,396.8,400.,397.6,392.1,400.8,400.6\}$                      | $\{397.3000,399.6000,401.0000,392.9000,396.8000,400.0000,397.6000,392.1000,400.8000,400.6000\}$ |
| x_:=mean(masses)  | 397.8700  |
| s_:=stDevPop(masses)  | 3.0480  |
| t:=nSolve(cdf(statdist("Studentt",{9}),x)=0.975,x)  | 2.2622  |
| $\left\{x_- - \frac{t \cdot s_-}{\sqrt{10-1}}, x_+ + \frac{t \cdot s_-}{\sqrt{10-1}}\right\}$ | $\{395.5717, 400.1683\}$  |
| t:=nSolve(cdf(statdist("Studentt",{9}),x)=0.995,x)  | 3.2498  |
| $\left\{x_- - \frac{t \cdot s_-}{\sqrt{10-1}}, x_+ + \frac{t \cdot s_-}{\sqrt{10-1}}\right\}$ | $\{394.5682, 401.1718\}$  |
| © Applying built-in function tInterval(...)   |   |
| tInterval masses  | Done  |

tPdf, cPdf and tInterval are built-in functions.

|   |  |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
|---|--|---------|--------------|----------|----------|----------|----------|------|----------|------|--------|------|--------|---------------|--------|-----|---------|
| $\{397.3000,399.6000,401.0000,392.9000,396.8000,400.0000,397.6000,392.1000,400.8000,400.6000\}$ |  |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| x_:=mean(masses)  | 397.8700   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| s_:=stDevPop(masses)  | 3.0480   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| t:=nSolve(cdf(statdist("Studentt",{9}),x)=0.975,x)  | 2.2622   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| $\left\{x_- - \frac{t \cdot s_-}{\sqrt{10-1}}, x_+ + \frac{t \cdot s_-}{\sqrt{10-1}}\right\}$   | $\{395.5717, 400.1683\}$   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| t:=nSolve(cdf(statdist("Studentt",{9}),x)=0.995,x)  | 3.2498   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| $\left\{x_- - \frac{t \cdot s_-}{\sqrt{10-1}}, x_+ + \frac{t \cdot s_-}{\sqrt{10-1}}\right\}$   | $\{394.5682, 401.1718\}$   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| © Applying built-in function tInterval(...)   |  |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| tInterval masses  | Done   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| stat.results  | <table border="1"> <tbody> <tr> <td>"Title"</td> <td>"t Interval"</td> </tr> <tr> <td>"CLower"</td> <td>395.5717</td> </tr> <tr> <td>"CUpper"</td> <td>400.1683</td> </tr> <tr> <td>"x̄"</td> <td>397.8700</td> </tr> <tr> <td>"ME"</td> <td>2.2983</td> </tr> <tr> <td>"df"</td> <td>9.0000</td> </tr> <tr> <td>"sx := sn-ix"</td> <td>3.2128</td> </tr> <tr> <td>"n"</td> <td>10.0000</td> </tr> </tbody> </table> | "Title" | "t Interval" | "CLower" | 395.5717 | "CUpper" | 400.1683 | "x̄" | 397.8700 | "ME" | 2.2983 | "df" | 9.0000 | "sx := sn-ix" | 3.2128 | "n" | 10.0000 |
| "Title"   | "t Interval"   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "CLower"  | 395.5717   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "CUpper"  | 400.1683   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "x̄"  | 397.8700   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "ME"  | 2.2983   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "df"  | 9.0000   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "sx := sn-ix"   | 3.2128   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| "n"   | 10.0000  |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| tInterval masses,1,0.99   | Done   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |
| {stat.CLower,stat.CUpper}   | $\{394.5682, 401.1718\}$   |         |              |          |          |          |          |      |          |      |        |      |        |               |        |     |         |

And this is the StudentT distribution for *DERIVE*:

$$\text{stud\_pdf}(x, n) := \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{(n \cdot \pi)} \cdot \Gamma\left(\frac{n}{2}\right)} \cdot \left(1 + \frac{x^2}{n}\right)^{-(n+1)/2}$$

$$\text{stud\_pdf}(x, 3) = \frac{6 \cdot \sqrt{3}}{\pi \cdot (x^2 + 3)^2}$$

$$\text{stud\_pdf}(2, 5) = 0.06509031032$$

$$\text{stud\_pdf}(-1, 10) = 0.2303619892$$

**The cdf is implemented as STUDENT(x,n).**

$$\text{stud\_cdf}(x, n) := \text{STUDENT}(x, n)$$

some simplifications:

$$\text{stud\_cdf}(x, 3) = \text{IF} \left[ x < 0, 1 - \text{STUDENT}(-x, 3), 1 - \left( \frac{\text{ACOT}\left(\frac{\sqrt{3} \cdot x}{3}\right)}{\pi} - \frac{\sqrt{3} \cdot x}{\pi \cdot (x^2 + 3)} \right) \right]$$

$$\text{stud\_cdf}(3, 3) = \frac{10 \cdot \pi + 3 \cdot \sqrt{3}}{12 \cdot \pi}$$

$$\text{stud\_cdf}(2, 5) = \frac{\text{ASIN}\left(\frac{2}{3}\right)}{\pi} + \frac{243 \cdot \pi + 148 \cdot \sqrt{5}}{486 \cdot \pi}$$

$$\text{stud\_cdf}(2, 5) = 0.9490302605$$

$$\text{stud\_cdf}(1.649, 10) = 0.9349182223$$

$$\text{stud\_cdf}(1.649, 10) = 0.9349182223$$

$$\text{NSOLVE}(\text{stud\_cdf}(x, 10) = 0.9349182223, x) = (x = 1.648999999)$$

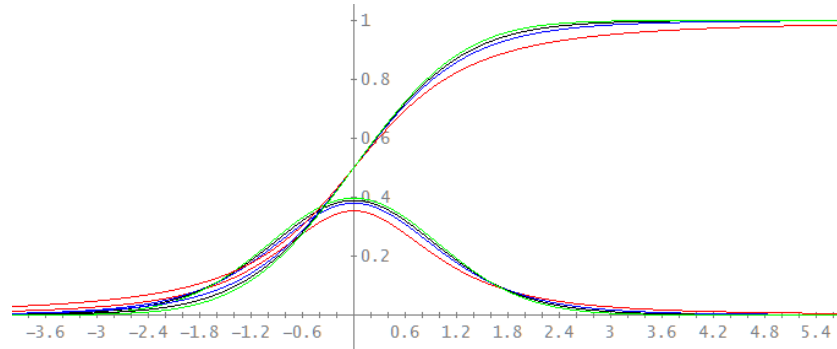
$$\text{NSOLVE}(\text{stud\_cdf}(x, 9) = 0.975, x) = (x = 2.262157127)$$

$$\text{NSOLVE}(\text{stud\_cdf}(x, 20) = 0.995, x) = (x = 2.8453397)$$

Next page shows the graphs of pdf and cdf for degrees of freedom 2, 5, 10, and 50 (red, blue, black and green):

$$[\text{stud\_pdf}(x, 2), \text{stud\_pdf}(x, 5), \text{stud\_pdf}(x, 10), \text{stud\_pdf}(x, 50)]$$

$$[\text{stud\_cdf}(x, 2), \text{stud\_cdf}(x, 5), \text{stud\_cdf}(x, 10), \text{stud\_cdf}(x, 50)]$$



Using Felipe Martinez Vargas'

`[ICDF_T(0.975, 9), ICDF_T(0.995, 9)]`

`[2.262157127, 3.249835520]`

<https://www.statisticshowto.com/probability-and-statistics/t-distribution/>

<https://www.scribbr.com/statistics/t-distribution/>

<https://www.scribbr.com/statistics/t-test/>

## Fratio distribution (Fisher distribution or Snedecor distribution)

Felipe de Jesus Martinez Vargas from Colima, Mexico sent a huge *Derive*-file treating great a number of distributions. I include the part with regard to the F-distribution (I leave it in Spanish, just for fun and to honor our many Spanish speaking DUG members, of course.):

### **Distribución F de Snedecor** (para una variable F)

La distribución F también es conocida como la distribución de relación de varianzas y tiene dos tipos de grados de libertad : grados de libertad del numerador y grados de libertad del denominador. Es la distribución de la relación de dos variables aleatorias independientes con distribuciones de chi-cuadrada, 37ad auna de las cuales se divide entre sus grados de libertad.

### **Snedecor's F distribution** (for an F variable)

The F-distribution is also known as the variance ratio distribution and has two types of degrees of freedom: degrees of freedom of the numerator and degrees of freedom of the denominator. It is the distribution of the ratio of two independent random variables with chi-square distributions, each of which is divided by its degrees of freedom.

Wikipedia: The *F*-distribution was developed by Fisher to study the behavior of two variances from random samples taken from two independent normal populations. In applied problems we may be interested in knowing whether the population variances are equal or not, based on the response of the random samples. Knowing the answer to such a question is also important in selecting the appropriate statistical methods to study their true means.

Density function: 
$$f_{m,n}(X) = \frac{\Gamma\left(\frac{m+n}{2}\right) \cdot m^{\frac{m}{2}} \cdot n^{\frac{n}{2}}}{\Gamma\left(\frac{m}{2}\right) \cdot \Gamma\left(\frac{n}{2}\right)} \frac{X^{\frac{m}{2}-1}}{(m+nX)^{\frac{m+n}{2}}}, \quad X > 0$$

Distribution function: 
$$F_{m,n}(x) = \frac{1}{B\left(\frac{m}{2}, \frac{n}{2}\right)} \int_0^{\frac{mx}{m+n}} t^{\frac{m}{2}-1} (1-t)^{\frac{n}{2}-1}, \quad x > 0$$

$m, n$  ... degrees of freedom

```

cdf(statdist("Fratio", {10 12}), 2)
.87249209
cdf(statdist("Fratio", {.03 .14}), .9)
.80457052
pdf(statdist("Fratio", {.03 .14}), .9)
.01321305
pdf(statdist("Fratio", {10 12}), 2)
.16708145
...tatdist("Fratio", {10,12}), 2)
MATH TOOL      RAD AUTO      FUNC 11/30

```

The Fratio-pdf is part of Bhuvanesh's math-tool library but the respective cdf is not. I added the cdf for V200 and TI-92 first and proceeded programming the functions for TI-Nspire and DERIVE as well.

Again, I start with *wxMaxima*:

```

(%i1) load(distrib)$
(%i2) pdf_f(0.9,0.03,0.14);
(%o2) 0.01321304674881406
(%i3) cdf_f(0.9,0.03,0.14);
(%o3) 0.8045705153064628
(%i4) pdf_f(4.5,15.0,20);
(%o4) 0.001557686795010424
(%i5) cdf_f(4.5,15.0,20);
(%o5) 0.9989082658061769

```

Two worked examples

In the course of a dust investigation in exhaust gases, it was suspected that the dust weight [mg] in part A of the pipe system varied more than in part B. <sup>[1]</sup>

Measurements showed:

|   |    |    |    |    |    |    |     |    |    |    |    |    |
|---|----|----|----|----|----|----|-----|----|----|----|----|----|
| A | 75 | 20 | 70 | 70 | 85 | 90 | 100 | 40 | 35 | 65 | 90 | 35 |
| B | 20 | 35 | 55 | 50 | 65 | 40 |     |    |    |    |    |    |

Is the difference between the variances significant ( $\alpha = 5\%$ )?

$$H_0: \sigma_A^2 = \sigma_B^2, H_a: \sigma_A^2 \neq \sigma_B^2$$

The test statistic is 
$$\frac{\sigma_A^2/m}{\sigma_B^2/n} = \frac{12 \cdot 11}{7625} \sim 2.67 \quad \text{with } h, m, n = 11, 5 \text{ (degrees of freedom)}$$

Comment: the greater  $\sigma^2$  forms the numerator of the test statistic

In the table of the book the value for  $F = 0.975$  (two-tailed test!) with  $m, n = 11, 5$  is not contained, but because of values nearby we can assume that its value is much greater than 2.67, i.e. we cannot reject  $H_0 \Rightarrow$  there is no difference.

Calculation with TI-Nspire: varSamp gives the variance of the sample, which is  $\sigma^2$  divided by sample size minus one (= degrees of freedom).

The screenshot shows a TI-Nspire calculator window with the following content:

```

1.1 1.2 1.3 *mathtool RAD
varSamp({ 75,20,70,70,85,90,100,40,35,65,90 })
679.356060606
varSamp({ 20,35,55,50,65,40 })
254.166666667
679.35606060608 2.67287630402
254.16666666666
nSolve(cdf(statdist("Fratio",{ 11,5 },x)=0.95)
4.70396723330

```

The annual salaries for a random sample of 16 actuaries working in California have a standard deviation of \$23,900. The annual salaries for a random sample of 17 actuaries working in New York have a standard deviation of \$18,800. Using this information, can you conclude that the standard deviation for actuaries is greater in in Cal than in NY? Use  $\alpha = 0.05\%$ .<sup>[2]</sup>

We will work with *DERIVE* now:

$$H_0: \sigma_A^2 \leq \sigma_B^2, H_a: \sigma_A^2 > \sigma_B^2 \text{ (claim)}$$

The test statistic:

$$\#1: \frac{23900^2}{18800^2} = 1.616144182$$

$$\#2: \text{NSOLVE}(\text{F\_DISTRIBUTION}(x, 15, 16) = 0.05, x) = (x = -6.160835682)$$

$$\#3: \text{NSOLVE}(\text{F\_DISTRIBUTION}(x, 15, 16) = 0.05, x, 0, 10) = (x = 2.352222753)$$

$$\#4: 2.352222753 > 1.616144182$$

Please note: In *DERIVE* we have to use the rejection region in the function. In our case is the critical value greater than the test statistic. So, we fail to reject  $H_0$ .

Additional comment: *DERIVE*'s function `VARIANCE([.....])` does the same job as Nspire's `varsamp`.

**Ejemplo (Felipe de Jesús) :**

En un laboratorio se efectuaron ciertas mediciones y se comprobó que seguían una distribución F con 10 grados de libertad en el numerador y 12 grados de libertad en el denominador.

Calcule el valor de 2.7534 del área bajo la curva de densidad.

**Example**

In a laboratory, certain measurements were made and found to follow an F distribution with 10 degrees of freedom in the numerator and 12 degrees of freedom in the denominator.

Calculate the value of 2.7534 of the area under the density curve.

Entering `Fdistribution(2.7534,10,12)` in Felipe's PDF Y CDF\_eng1.dfw gives all important parameters of this distribution:

```
FDistribution(2.7534, 10, 12)
```

|                                |               |
|--------------------------------|---------------|
| ◀ PARAMETERS ▶                 | ┆-----┆       |
| Numerator Degrees of Freedom   | 10            |
| Denominator Degrees of Freedom | 12            |
| ◀ PROBABILITY ▶                | ┆-----┆       |
| Value of Variable (f)          | 2.7534        |
| P(F = f) =                     | 0.058648501   |
| P(F ≤ f) =                     | 0.9500007763  |
| P(F ≥ f) =                     | 0.1086477246  |
| P(F < f) =                     | 0.8913522753  |
| P(F > f) =                     | 0.04999922362 |
| Mean                           | 1.2           |
| Median                         | ...           |
| Mode                           | 0.6857142857  |
| SDeviation                     | 0.8485281374  |
| Variance                       | 0.72          |
| Skewness                       | 2.828427124   |
| Kurtosis                       | 21            |

```
ICDF_F(Prob, v1, v2) := RHS(NSOLVE(DISTR_F_N(x, v1, v2, 1) = Prob, x, 0, ∞))
```

```
ICDF_F(0.9500007756, 10, 12)
```

```
2.753399986
```

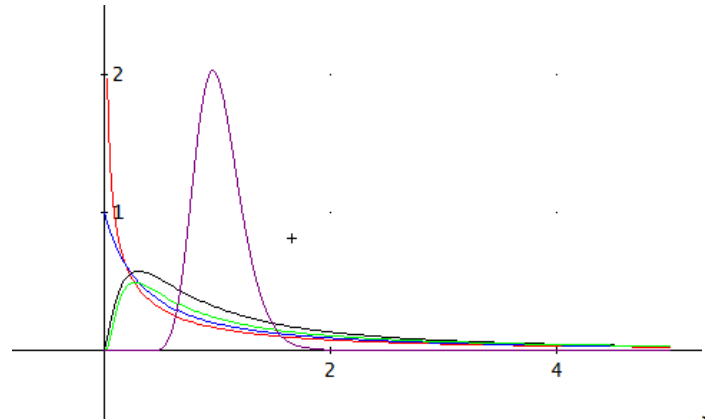
(from PDF Y CDF\_eng1.dfw)

Next page presents the F-distribution table followed by some pdf- and cdf-graphs:



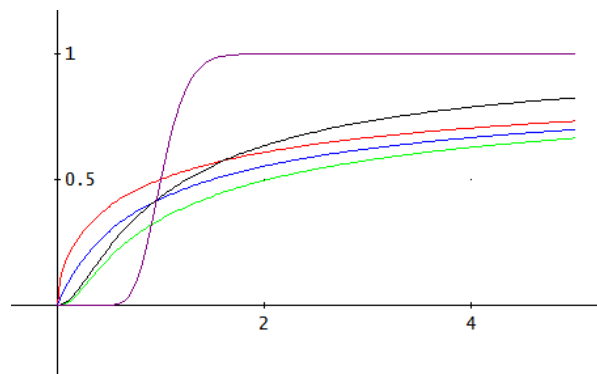
$$F\_PDF(v1, v2, Left, Right) := VECTOR \left( [f, DISTR\_F\_N(f, v1, v2, 0)], f, Left, Right, \frac{Right - Left}{200} \right)$$

$$\begin{bmatrix} F\_PDF(1, 1, 0, 5) \\ F\_PDF(2, 1, 0, 5) \\ F\_PDF(5, 2, 0, 5) \\ F\_PDF(10, 1, 0, 5) \\ F\_PDF(100, 100, 0, 5) \end{bmatrix}$$



$$F\_CDF(v1, v2, Left, Right) := VECTOR \left( [f, DISTR\_F\_N(f, v1, v2, 1)], f, Left, Right, \frac{Right - Left}{200} \right)$$

$$\begin{bmatrix} F\_CDF(1, 1, 0, 5) \\ F\_CDF(2, 1, 0, 5) \\ F\_CDF(5, 2, 0, 5) \\ F\_CDF(10, 1, 0, 5) \\ F\_CDF(100, 100, 0, 5) \end{bmatrix}$$



My “selfmade” F-distribution functions for *DERIVE* are following:

See first the density function:

$$\begin{aligned} f\_pdf(x, m, n) &:= \\ \text{#7:} \quad & \begin{cases} \text{If } x \leq 0 \\ 0 \\ \Gamma((n+m)/2) \cdot m^{(m/2)} \cdot n^{(n/2)} / (\Gamma(n/2) \cdot \Gamma(m/2)) \cdot x^{(m/2 - 1)} / (m \cdot x + \\ \\ n)^{((m+n)/2)} \end{cases} \end{aligned}$$

$$\text{\#8:} \quad \text{regincbeta}(a, b, x) := - \frac{1 \cdot (1-x)^b \cdot \Gamma(a+b)}{\Gamma(b+1) \cdot \Gamma(a)} + \frac{x^a \cdot \Gamma(a+b)}{\Gamma(a+1) \cdot \Gamma(b)} +$$

$$\frac{\Gamma(a+b)}{\Gamma(b+1) \cdot \Gamma(a)} \cdot (1-b) \cdot \int_0^x \frac{(1-t)^b \cdot t^a \cdot (t-1) + t^a \cdot (t-1)}{t \cdot (t-1)} dt$$

What concerns `regincbeta` (regularized incomplete Beta function) see page 45.

And this is the distribution function:

$$\#9: f\_cdf(x, m, n) := \text{regincbeta}\left(\frac{m}{2}, \frac{n}{2}, \frac{m \cdot x}{m \cdot x + n}\right)$$

$$\#10: \text{APPROX}(f\_cdf(2, 10, 12)) = 0.8724920917$$

$$\#11: \text{APPROX}(f\_pdf(2, 10, 12)) = 0.1670814526$$

$$\#12: \text{NSOLVE}(f\_cdf(x, 10, 12) = 0.95, x)$$

$$\#13: x = -6.615400165$$

$$\#14: \text{NSOLVE}(f\_cdf(x, 10, 12) = 0.95, x, 0, 10)$$

$$\#15: x = 2.753386795$$

$$\#16: \text{NSOLVE}(f\_cdf(x, 7, 7) = 0.95, x, 0, 10)$$

$$\#17: x = 3.787043541$$

The TI-Nspire implementation is as follows:

The screenshot shows the TI-Nspire calculator interface. On the left, a list of function calls and their results is displayed:

- $pdf(\text{statdist}(\text{"Fratio"}, \{0.03, 0.14\}), 0.9)$  → 0.013213047066
- $cdf(\text{statdist}(\text{"Fratio"}, \{0.03, 0.14\}), 0.9)$  → 0.804570552935
- $pdf(\text{statdist}(\text{"Fratio"}, \{10, 12\}), 2)$  → 0.167081452673
- $cdf(\text{statdist}(\text{"Fratio"}, \{10, 12\}), 2.7534)$  → 0.950000775989
- $nSolve(cdf(\text{statdist}(\text{"Fratio"}, \{10, 12\}), x) = 0.95, x)$  → 2.75338676889
- $pdf(\text{statdist}(\text{"Fratio"}, \{15, 20\}), 4.5)$  → 0.001557686795
- $cdf(\text{statdist}(\text{"Fratio"}, \{15, 20\}), 4.5)$  → 0.998908246100
- $varSamp(\{6.9, 7.7, 4.11, 5.17, 9.11, 9.10, 2.7, 8\})$  → 13.6114285714
- $varSamp(\{6.4, 8.3, 7.9, 8.8, 10.1, 11.5, 8.7, 9.7\})$  → 2.35642857143

On the right, the function definitions are shown in a code editor:

```
pdf
1/49
If distype="fratio" Then
  If pars[1]≤0 or pars[2]≤0
    Return "Error: wrong degrees of freedom"
  If xx<0
    Return "only positive arguments"
  Return  $\frac{\frac{pars[1]}{2} \cdot \frac{pars[2]}{2} \cdot \frac{pars[1]}{xx}}{\text{mathtool}\beta\left(\frac{p}{2}, \frac{p}{2}, \frac{p}{pars[1]}\right)}$ 

cdf
6/53
EndIf
If distype="fratio" Then
  If pars[1]≤0 or pars[2]≤0
    Return "Error: wrong degrees of freedom"
  Return  $\text{mathtool}\text{regincbeta}\left(\frac{pars[1]}{2}, \frac{pars[2]}{2}, \frac{pa}{pars[1]}\right)$ 
EndIf
If distype="studentt" Then
  If pars[1]≤0
    Return "Error: wrong dregree of freedom"
```

It is nice to program the functions and to be satisfied that they work as expected. But we must not forget, that the F-distribution is also represented in TI-Nspire's CATALOG:

FPdf(Xval,dfNumer,dfDenom), FCdf(low bound, up bound, dfNumer, dfDenom) and invF(Area, dfNumer, dfDenom) are functions implemented in TI\_Nspire.

I used them as references for my functions which are part of the mathtools-library based on Bhuvanesh Bhatt's idea.

|   |                |
|---|----------------|
| 2.3564285714286                                     |                |
| $nSolve(cdf(statdist("Fratio",\{7,7\}),x)=0.95,x)$  | 3.78704353993  |
| $invF(0.95,7,7)$                                    | 3.78704353997  |
| $FPdf(2,10,12)$                                     | 0.167081452673 |
| $FCdf(0,4.5,15,20)$                                 | 0.998908265808 |
| $nSolve(cdf(statdist("Fratio",\{11,5\}),x)=0.95,x)$ | 4.70396723330  |
| $invF(0.95,11,5)$                                   | 4.70396723325  |
| $varSamp(\{75,20,70,70,85,90,100,40,35,65,90,35\})$ | 679.356060606  |
| $varSamp(\{20,35,55,50,65,40\})$                    | 254.166666667  |
| $\frac{679.35606060608}{254.16666666666}$           | 2.67287630402  |
| $nSolve(cdf(statdist("Fratio",\{11,5\}),x)=0.95,x)$ | 4.70396723330  |

pdf 1/49

```

If disttype="fratio" Then
  If pars[1]≤0 or pars[2]≤0
    Return "Error: wrong degrees of freedom"
  If xx<0
    Return "only positive arguments"
  Return  $\frac{pars[1]^{pars[1]} \cdot pars[2]^{pars[2]} \cdot xx^{-pars[1]-pars[2]}}{\text{mathtool}\beta\left(\frac{pars[1]}{2}, \frac{pars[2]}{2}, \frac{pars[1]}{pars[1]+pars[2]}\right)}$ 

```

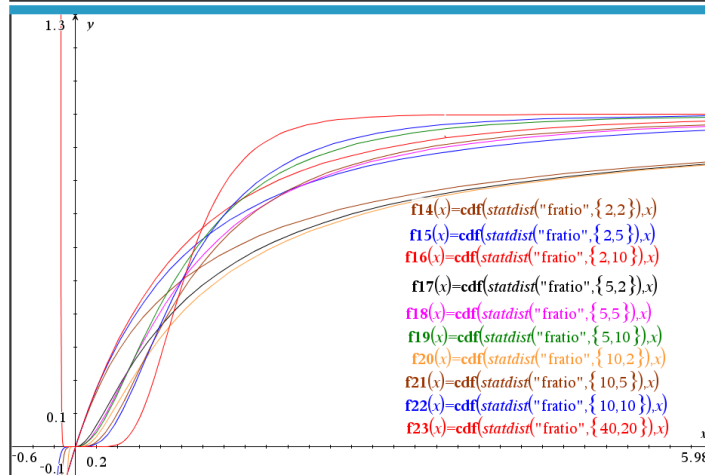
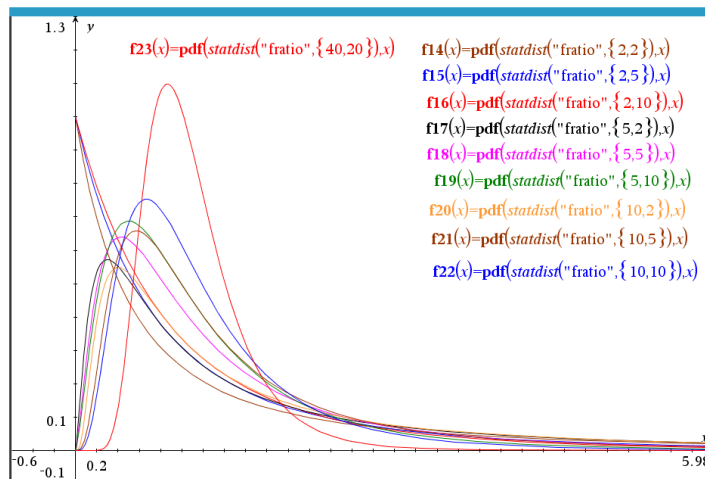
---

cdf 6/53

```

EndIf
If disttype="fratio" Then
  If pars[1]≤0 or pars[2]≤0
    Return "Error: wrong degrees of freedom"
  Return  $\text{mathtool}\text{regincbeta}\left(\frac{pars[1]}{2}, \frac{pars[2]}{2}, \frac{pars[1]}{pars[1]+pars[2]}\right)$ 
EndIf
If disttype="studentt" Then
  If pars[1]≤0
    Return "Error: wrong degree of freedom"

```



Useful Links:

<https://www.sciencedirect.com/topics/mathematics/f-distribution>

<https://www.statisticshowto.com/probability-and-statistics/hypothesis-testing/f-test/>

<https://mathepedia.de/F-Verteilung.html>

Referencies:

[1] *E. Kreyszig, Statistische Methoden und ihre Anwendungen*

[2] *Larson & Farber, Elementary Statistics Picturing the World*

## Beta Distribution & generalized Beta Distribution

The beta distribution has been applied to model the behavior of random variables limited to intervals of finite length in a wide variety of disciplines. It is a probability distribution of probabilities. It is useful as apriori distribution for unknown probabilities in Bayesian models and in PERT-analysis. On the first glance it looks similar to the binomial distribution. While The binomial distribution models the number of successes, the Beta distribution models the probabilities of successes.

The parameters  $p$  and  $q$  can be interpreted as number of successes and number of failures. They determine the shape of the function.

Standard form:

$$\text{Density function: } f_{p,q}(X) = \begin{cases} \frac{1}{B(p,q)} \cdot X^{p-1} \cdot (1-X)^{q-1}, & 0 \leq X \leq 1 \\ 0 & \text{else} \end{cases}$$

$$\text{Distribution function: } F_{p,q}(x) = \begin{cases} 0 & x < 0 \\ \frac{1}{B(p,q)} \int_0^x u^{p-1} \cdot (1-u)^{q-1} du = \text{Incomplete\_Beta}(x, p, q) & 0 \leq x \leq 1 \\ 1 & x > 1 \end{cases}$$

The four parameters Beta distribution (generalized Beta distribution) extends the interval for  $a \leq x \leq b$ :

$$\text{Density function: } f_{a,b,p,q}(X) = \begin{cases} \frac{1}{B(p,q)} \cdot \frac{(X-a)^{p-1} \cdot (b-X)^{q-1}}{(b-a)^{p+q-1}}, & a \leq X \leq b \\ 0 & \text{else} \end{cases}$$

Distribution function:

$$F_{a,b,p,q}(x) = \begin{cases} 0 & x < a \\ \frac{1}{B(p,q) \cdot ((b-a)^{p+q-1})} \int_a^x (u-a)^{p-1} \cdot (b-u)^{q-1} du = \text{Incomplete\_Beta}\left(\frac{x-a}{b-a}, p, q\right) & a \leq x \leq b \\ 1 & x > b \end{cases}$$

**Example** (from Felipe's PDF Y CDF\_eng1.dfw)

In the household budget, the portion devoted to health follows a Beta distribution(2,2).

What is the probability that more than 25% of the family budget is spent on health?

$$1 - \text{DISTR\_BETA\_N}(0.25, 2, 2, 1) = 0.84375$$

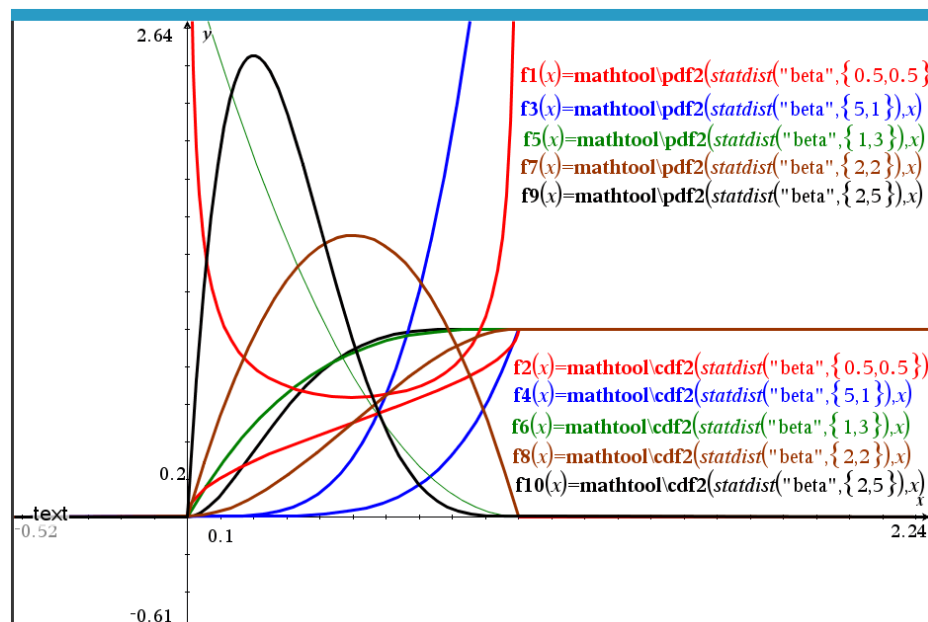
|   |                |
|---|----------------|
| <code>mathtool\pdf2(statdist("Beta",{0.5,0.5}),0.5)</code>        | 0.636619772368 |
| <code>mathtool\pdf2(statdist("Beta",{0.5,0.5}),0.2)</code>        | 0.795774715459 |
| <code>mathtool\cdf2(statdist("Beta",{0.5,0.5}),0.5)</code>        | 0.500000000000 |
| <code>mathtool\cdf2(statdist("Beta",{0.5,0.5}),0.2)</code>        | 0.295167235301 |
| <code>mathtool\pdf2(statdist("Beta",{0.5,0.5}),0.2)</code>        | 0.795774715459 |
| <code>mathtool\pdf2(statdist("GenBeta",{0.5,0.5,0,1}),0.2)</code> | 0.795774715459 |
| <code>mathtool\cdf2(statdist("GenBeta",{0.5,0.5,0,1}),0.2)</code> | 0.295167235301 |
| <code>mathtool\cdf2(statdist("GenBeta",{3,1,2,5}),3)</code>       | 0.037037037037 |

```

"pdf2" stored successfully
EndIf
If disttype="genbeta" Then
aa:=pars[1]:bb:=pars[2]:c:=pars[3]:d:=pars[4]
If d<=c:Return "Wrong Domain"
If real(aa)≠0 and real(bb)≠0 and imag(aa)=0 and im
Return {
0,
1
mathtool\beta(aa,bb)·(d-c)aa+bb-1 (xx-
0,
}
EndIf

"cdf2" stored successfully
EndIf
If disttype="beta" Then
aa:=pars[1]:bb:=pars[2]
If real(aa)≠0 and real(bb)≠0 and imag(aa)=0 and im
Return {
0, xx≤0
mathtool\regincbeta(aa,bb,xx),0<xx<1
1,
}
EndIf
If disttype="genbeta" Then
aa:=pars[1]:bb:=pars[2]:c:=pars[3]:d:=pars[4]
If d<=c:Return "Wrong Domain"

```



Pdf and cdf with self-made *DERIVE* functions:

```
b_pdf(p, q, x, a := 0, b := 1) :=
  If x ≥ a ^ x ≤ b
    1/EULER_BETA(p, q)·(x - a)^(p - 1)·(b - x)^(q - 1)/(b - a)^(p + q - 1)
  0
```

```
b_cdf(p, q, x, a := 0, b := 1) :=
  If x ≤ a
    0
  If x ≤ b
    INCOMPLETE_BETA((x - a)/(b - a), p, q)
  1
```

$b\_pdf(0.5, 0.5, 0.5) = 0.6366197723$

$b\_pdf(0.5, 0.5, 0.2) = 0.7957747154$

$b\_pdf(0.5, 0.5, 0.1) = 1.061032953$

$b\_pdf(0.5, 0.5, 2.1, 2, 3) = 1.061032953$

$b\_cdf(0.5, 0.5, 0.5) = \frac{1}{2}$

$b\_cdf(0.5, 0.5, 0.2) = 0.2951672353$

$b\_cdf(2, 5, 4, 2, 8) = 0.6488340192$

$\int_2^4 b\_pdf(2, 5, x, 2, 8) dx = 0.6488340192$

$\int_2^8 b\_pdf(2, 5, x, 2, 8) dx = 1$

<https://ichi.pro/de/beta-verteilung-intuition-beispiele-und-ableitung-227603014965167>

(in German only, very informative)

[https://www-docs.b-tu.de/fg-kommunikationstechnik/public/papers/2014\\_Wolff\\_ZurStatistischenAuswertungVonErkennungsergebnissen.pdf](https://www-docs.b-tu.de/fg-kommunikationstechnik/public/papers/2014_Wolff_ZurStatistischenAuswertungVonErkennungsergebnissen.pdf)

<https://towardsdatascience.com/beta-distribution-intuition-examples-and-derivation-cf00f4db57af>

<https://vitalflux.com/beta-distribution-explained-with-python-examples/>

[http://varianceexplained.org/statistics/beta\\_distribution\\_and\\_baseball/](http://varianceexplained.org/statistics/beta_distribution_and_baseball/)

[https://wiki.analytica.com/index.php?title=Beta\\_distribution](https://wiki.analytica.com/index.php?title=Beta_distribution)

## Beta-, incomplete & regularized incomplete Beta function

These functions are needed for several probability distributions.

I found two definitions for the Beta function:

$$B(x, y) = \frac{\Gamma(x) \cdot \Gamma(y)}{\Gamma(x + y)}, \quad B(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt, \quad x, y \text{ complex}$$

The screenshot shows a CAS interface with two panes. The left pane displays the following results:

- $\text{beta}(5,3) = \frac{1}{105}$
- $\text{beta}(5,3) = 0.009523809524$
- $\text{beta}(5+i, 3-2 \cdot i)$
- $\cos\left(\frac{\ln(733)}{2} - \frac{2 \cdot \ln(2) + 2 \cdot \tan^{-1}\left(\frac{121359100408574310}{554854015910717935}\right)}{2}\right)$
- $\text{beta}(5+i, 3-2 \cdot i) = 0.000266126972 + 0.004409580407 \cdot i$
- $\int_0^1 t^{x-1} \cdot (1-t)^{y-1} dt |_{x=5 \text{ and } y=3} = \frac{1}{105}$
- $\int_0^1 t^{x-1} \cdot (1-t)^{y-1} dt |_{x=5+i \text{ and } y=3-2 \cdot i} = 0.000266127055 + 0.004409580755 \cdot i$

The right pane shows a code editor with the following content:

```

beta
2/2
Define LibPub beta(p,q)=
Func
© Bhuvanesh Bhatt
when(q=1-p, pi * sin(pi * p),
mathtool\gamma(p) * mathtool\
mathtool\gamma(p+q),
EndFunc

```

Below the code editor, the same mathematical expressions as in the left pane are displayed, including the gamma function ratio and the complex beta function result.

Beta function in *DERIVE*:

$$\text{EULER\_BETA}(5, 3) = \frac{1}{105}$$

$$\text{EULER\_BETA}(5 + i, 3 - 2 \cdot i) = \frac{\left(-\frac{547}{204425} + \frac{2579 \cdot i}{204425}\right) \cdot (1 + i)! \cdot (1 - 2 \cdot i)!}{(1 - i)!}$$

$$\text{EULER\_BETA}(5 + i, 3 - 2 \cdot i) = 0.0002661270550 + 0.004409580753 \cdot i$$

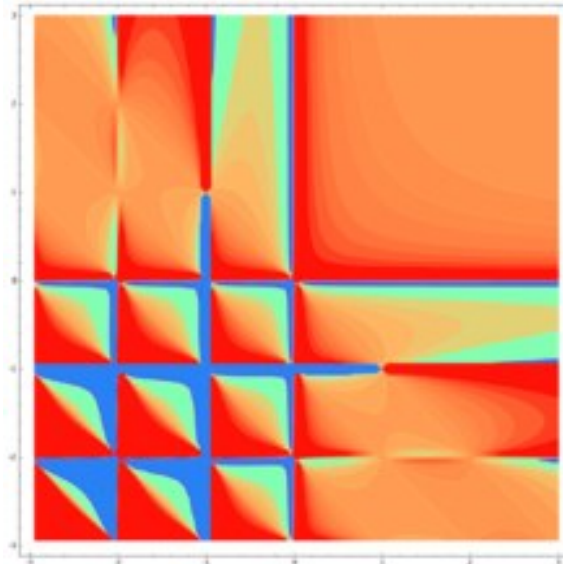
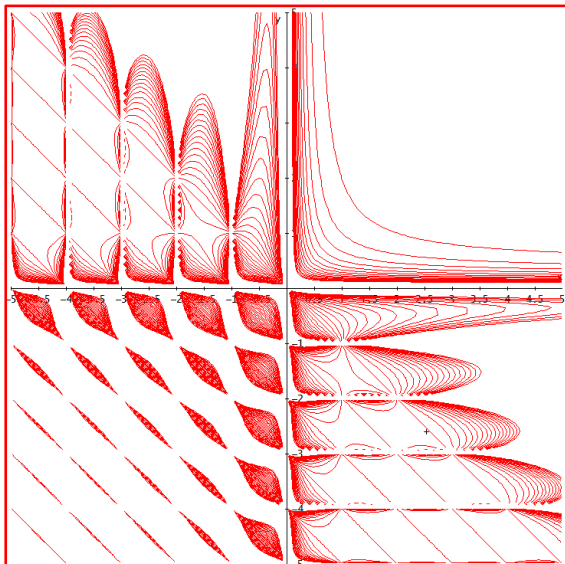
$$\frac{\Gamma(5) \cdot \Gamma(3)}{\Gamma(5 + 3)} = \frac{1}{105}$$

$$\frac{\Gamma(5 + i) \cdot \Gamma(3 - 2 \cdot i)}{\Gamma(5 + i + 3 - 2 \cdot i)} = 0.0002661270550 + 0.004409580753 \cdot i$$

$$\int_0^1 t^{5-1} \cdot (1-t)^{3-1} dt = \frac{1}{105}$$

$$\int_0^1 t^{5+i-1} \cdot (1-t)^{3-2 \cdot i-1} dt = 0.0002661270588 + 0.004409580840 \cdot i$$

Pretty plots based on the Beta function:



VECTOR(z = EULER\_BETA(x, y), z, -8, 8, 0.5)

Contour plot of the Beta function (*DERIVE* left and Wikipedia right)

The **incomplete Beta function** is defined as:  $B(x; a, b) = \int_0^x t^{a-1} (1-t)^{b-1} dt$ .

And the **regularized incomplete Beta function** is defined as:  $I_x(a, b) = \frac{B(x; a, b)}{B(a, b)}$ .

This function is the cumulative distribution function of the Beta distribution.

|  |                                      |   |     |
|--|--------------------------------------|---|-----|
| $incbeta\left(1, \frac{-3}{2}, 2\right)$                     | $\frac{4}{3}$                        | incbeta   | 5/8 |
| $incbeta(5, 4, 9)$   | $\frac{1142738125}{396}$             | Define LibPub <b>incbeta</b> (z,a,b)=               |     |
| $incbeta\left(\frac{1}{2}, 2, \frac{-1}{2}\right)$           | $3 \cdot \sqrt{2} - 4$               | Func  |     |
| $incbeta\left(\frac{-1}{2}, \frac{1}{2}, 2\right)$           | $\frac{7 \cdot \sqrt{2}}{6} \cdot i$ | © Bhuvanesh Bhatt                                   |     |
| $regbeta\left(1, \frac{-3}{2}, 2\right)$                     | 1                                    | If when(z=0 and real(a)≠0, true, false, false)      |     |
| ©  |                                      | Return when(real(a)>0, 0, undef)                    |     |
| $regbeta\left(\frac{-1}{2}, \frac{1}{2}, 2\right)$           | $\frac{7 \cdot \sqrt{2}}{8} \cdot i$ | If when(z=1 and real(b)>0, true, false, false)      |     |
| $regbeta(5, 4, 9)$   | $\frac{5713690625}{16}$              | Return $\text{mathtool}\backslash\text{beta}(a, b)$ |     |
| $regbeta\left(\frac{3}{2}, 2, 4\right)$                      | $\frac{9}{16}$                       | If when(real(a)>0, true, false, false)              |     |
| $regbeta\left(\frac{-3}{2}, \frac{5}{2}, \frac{9}{2}\right)$ | $721.321665304 \cdot i$              | Return $\int_0^z (t^{a-1} \cdot (1-t)^{b-1}) dt$    |     |
|  |                                      | Return <i>unknown</i>                               |     |
|  |                                      | EndFunc   |     |
|  |                                      | regbeta   | 2/2 |
|  |                                      | Define LibPub <b>regbeta</b> (z,a,b)=               |     |
|  |                                      | Func  |     |
|  |                                      | © _JB   |     |
|  |                                      | $\text{mathtool}\backslash\text{incbeta}(z, a, b)$  |     |
|  |                                      | $\text{mathtool}\backslash\text{beta}(a, b)$        |     |
|  |                                      | EndFunc   |     |



This function is the cumulative distribution function of the Beta distribution. Designation of the functions is not unique. *DERIVE*'s `INCOMPLETE_BETA` is the regularized beta function:

$$\text{INCOMPLETE\_BETA}\left(-\frac{1}{2}, \frac{1}{2}, 2\right) = \frac{7 \cdot \sqrt{2} \cdot i}{8}$$

$$\text{INCOMPLETE\_BETA}(1.5, 2, 4) = \frac{9}{16}$$

$$\text{INCOMPLETE\_BETA}(-1.5, 2.5, 4.5) = i \cdot \left( \frac{\text{LN}(\sqrt{15} + 4)}{\pi} + \frac{4092 \cdot \sqrt{15}}{7 \cdot \pi} \right)$$

$$\text{INCOMPLETE\_BETA}(-1.5, 2.5, 4.5) = 721.3216653 \cdot i$$

This is *WxMaxima*:

```
(%i1) beta_incomplete(-3/2,2,1);
(%o1) 4/3
(%i2) beta_incomplete_regularized(-3/2,2,1);
(%o2) 1
(%i3) beta_incomplete(4,9,5);
WARNING: redefining MAXIMA::SIMP-UNIT-STEP in DEFUN
WARNING: redefining MAXIMA::SIMP-POCHHAMMER in DEFT
(%o3) 1142738125/396
(%i4) beta_incomplete_regularized(4,9,5);
(%o4) 5713690625
(%i13) beta_incomplete(1/2,2,-1/2);
(%o13) 7 %i / (3 * sqrt(2))
(%i14) beta_incomplete_regularized(1/2,2,-1/2);
(%o14) 7 %i / 2^(5/2)
(%i11) beta_incomplete(2.5,4.5,-1.5);
(%o11) 15.4909100704717 %i + 4.742723358420668 10^-15
(%i12) beta_incomplete_regularized(2.5,4.5,-1.5);
(%o12) 721.3216653041629 %i + 2.208410671425955 10^-13
(%i16) beta_incomplete_regularized(2,4,3/2);
(%o16) 9/16
*
```

Take care of the order of the arguments.

```
Define LibPub regincbeta(a,b,x)=
Func
© JB according DERIVE-function


$$\frac{-(1-x)^b \cdot \text{mathtool}\backslash\text{gamma}(a+b)}{\text{mathtool}\backslash\text{gamma}(b+1) \cdot \text{mathtool}\backslash\text{gamma}(a)} + \frac{x^a \cdot \text{mathtool}\backslash\text{gamma}(a+b)}{\text{mathtool}\backslash\text{gamma}(a+1) \cdot \text{mathtool}\backslash\text{gamma}(b)} + \frac{\text{mathtool}\backslash\text{ga}}{\text{mathtool}\backslash\text{gamma}(b+1)}$$


EndFunc
```

**Frohe Weihnachten und viel Erfolg und Gesundheit  
im Neuen Jahr 2022**

**Merry Christmas and much Success and  
Health for New Year 2022**



Christ Roses and Winter Time in Upper Austria

