THE DERIVE - NEWSLETTER #122

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE

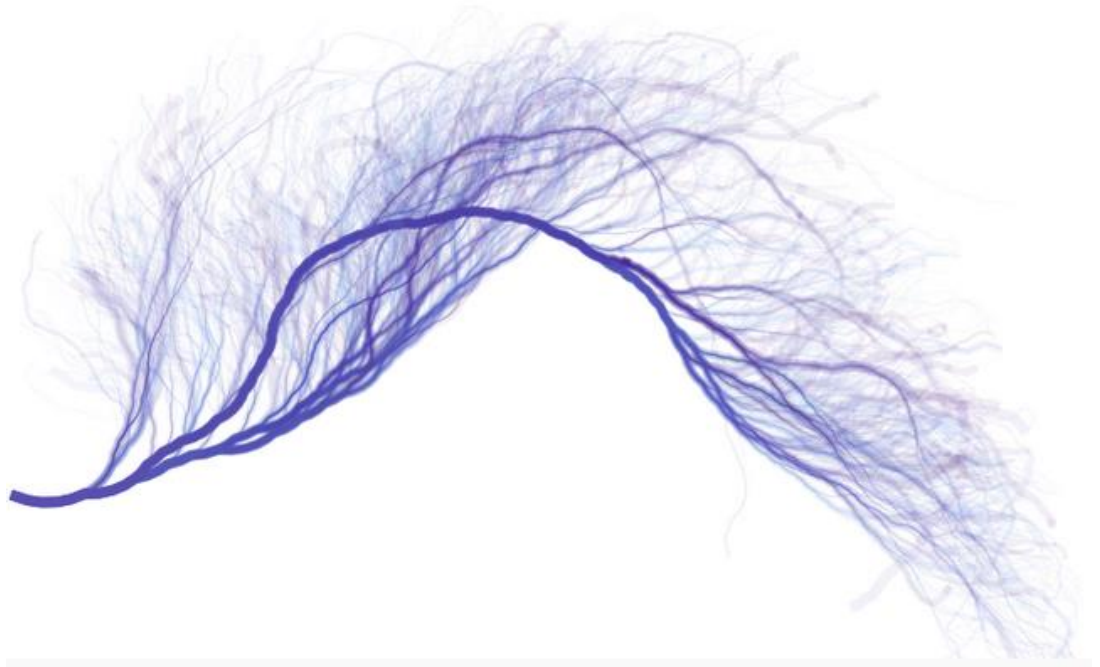
USER GROUP

+ CAS-TI

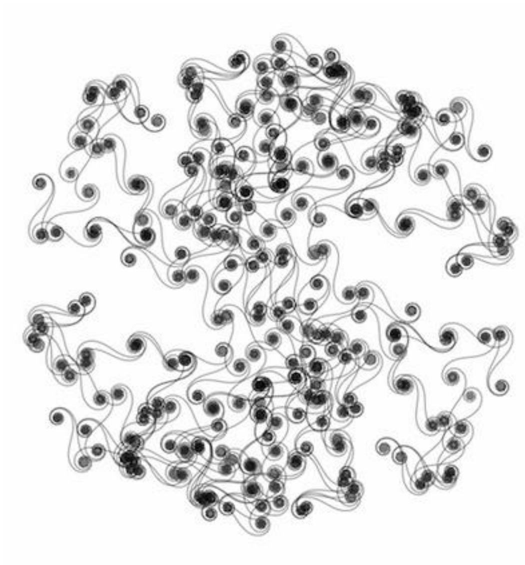
Contents:

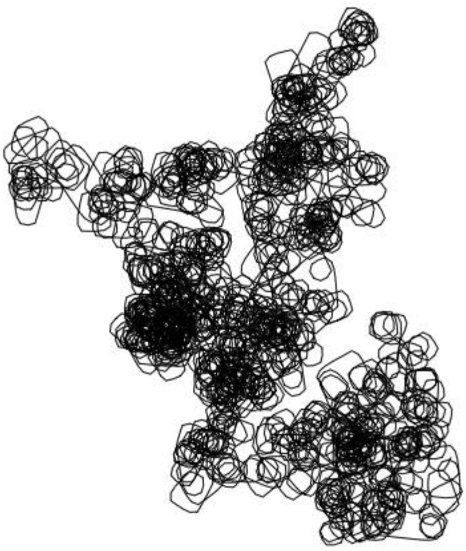
June 2021

When I did some internet research connected with the Collatz- or 3N+1-sequence I discovered these remarkable images.



https://community.wolfram.com/groups/-/m/t/558256





https://community.wolfram.com/groups/-/m/t/554194

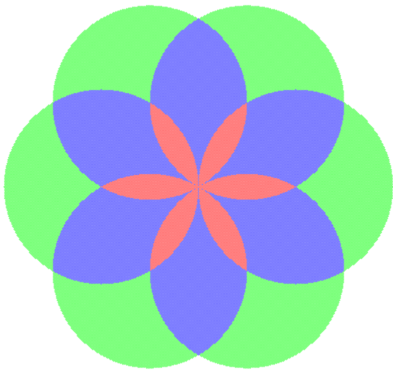
https://demonstrations.wolfram.com/CurlicueFractal/

https://reference.wolfram.com/language/ref/AnglePath.html

Dear DUG-Members,

I would like to greet you warmly and at the same time I am pleased to be able to present the DNL122. In this issue, there are few but extensive contributions, some of which refer to previous DNLs. So Matthew Myers has programmed more functions for the only "computer" at his disposal - the TI-89. . The Collatz sequence and the Vigenere code have been covered before. It may be appealing to compare the program styles. I found his function *extract* interesting from the programming point of view, while *is_pr* encouraged me to make another excursion into number theory. I have learnt a lot about “primitive roots”. His *polygon* for the TI was a welcome opportunity for me to use TI Python programming the TI-Nspire's graphics - although I'm really no Python expert.
All functions for TI-89/92/V200 and for TI-Nspire as well are contained in MTH122.zip.

Michel Beaudin has sent a longer article in which, as he himself writes, he summarizes some known and less known facts about equations, systems of equations, etc. I have left this in its original (French) version. I hope that you can follow, but I have the intention to produce an English version during summer.



Then there is a very extensive overview of various probability distributions, to which I was inspired by three functions from Bhuvanesh Bhatt's mathtool library. This was an exciting journey of discovery for me, and it's far from over. There is a long list of more distributions left to be presented. This could also be an interesting field of activity for students. mathtool.tns should be stored in the MyLib-folder of TI-Nspire, as well as mm.tns for the Matthew-Myers-functions.

Finally, the interesting problem from the Bolyai competition should be mentioned. Arguing, simulating, programming complement each other in a nice way. Thanks to Hubert and Wilfried.

I remain with best regards

Josef

Liebe DUG-Mitglieder,

ich möchte Euch herzlich grüßen und freue mich gleichzeitig den DNL122 vorlegen zu können. In dieser Ausgabe gibt es wenige, aber dafür umfangreiche Beiträge, die zum Teil mit früheren DNLs in Beziehung stehen. So hat Matthew Myers weitere Funktionen für den einzigen „Computer“, der ihm zur Verfügung steht – den TI-89 – programmiert. Die Collatz-Folge und der Vigenere-Code wurden schon früher behandelt. Es mag reizvoll sein, die Programmstile zu vergleichen. Seine Funktion *extract* habe ich vom Programm her interessant gefunden, während mich *is_pr* zu einem weiteren Ausflug in die Zahlentheorie ermuntert hat. Ich habe einiges über „primitive Wurzeln“ gelernt. Sein *polygon* für den TI gab die Möglichkeit, mit TI-Python auch die Grafik des TI-Nspire zu programmieren – obwohl ich wirklich kein Python-Experte bin.
Alle Funktionen für TI-89/92/V200 als auch für TI-Nspire sind in MTH122-zip enthalten.

Michel Beaudin hat einen längeren Artikel geschickt, in dem er, wie er selbst schreibt, Bekanntes und weniger Bekanntes zusammenfasst. Ich habe das einmal in seiner Originalfassung (Französisch) belassen. Ich schicke Euch aber mit dieser Info eine deutsche Übersetzung als attachment mit. Ich hoffe, dass sie halbwegs gelungen ist.

Dann gibt es noch einen sehr umfangreichen Überblick über diverse Wahrscheinlichkeitsverteilungen, zu der ich von drei Funktionen aus der mathtool-Bibliothek von Bhuvanesh Bhatt angeregt wurde. Das war für mich eine spannende Entdeckungsreise, die noch lange nicht zu Ende ist. Es gibt ja eine lange Liste von Verteilungen. Das könnte auch für Schülerinnen und Schüler ein interessantes Betätigungsfeld sein. mathtool.tns sollte im MyLib-folder von TI-Nspire gespeichert werden, ebenso wie mm.tns für die Matthew-Myers-Funktionen.

Schließlich soll auch das interessante Problem aus dem Bolyai-Wettbewerb erwähnt werden. Argumentieren, Simulieren, Programmieren ergänzen einander auf schöne Weise. Dank an Hubert und Wilfried.

Ich verbleibe mit besten Grüßen

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: September 2021

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
- Statistics of Shuffling Cards, H. Ludwig, 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, n-Circle Figures, 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
- Double-Die-Encryption – Doppelwürfelverschlüsselung
- TaxiCab Conics, Two alternate Approaches to Conics, R. Haas, USA
- QR-Code light, 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

More Functions for the Handheld TIs

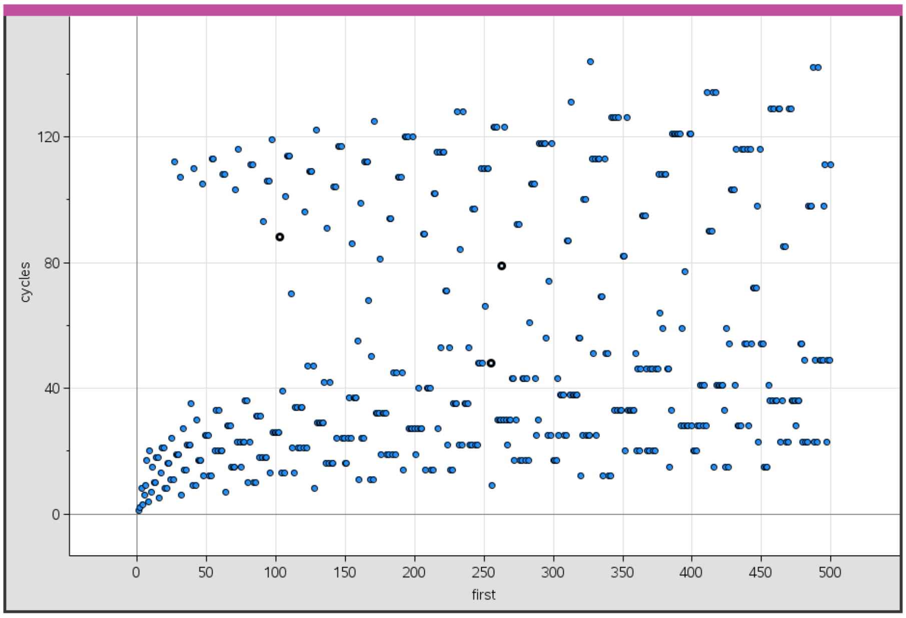
Matthew Myers, USA

Matthew Myers sent a new bundle of TI-handheld (TI-89, TI-92 and Voyage 200) programs via his mentor Robert Haas. All files are contained in MTH122.zip. I present his functions as Nspire-procedures which are part of the library mm.tns which is to be stored in the TI-NspireCX\MyLib-folder.
I will not reprint the function codes but present the results on screen shots.

look_say(3,6)	{3,13,1113,3113,132113,1113122113,311311222113}
look_say(2,7)	{2,12,1112,3112,132112,1113122112,311311222112,13211321322112}
look_say(4,5)	{4,14,1114,3114,132114,1113122114}
in_one({1,2,3,4,5,10},{1,2,4,6,7})	{3,5,10,6,7}
in_one({*ab*,*ac*,*ad*,*ae*},{*ae*,*ab*,*bc*})	{*ac*,*ad*,*bc*}
in_both({1,2,3,4,5,10},{1,2,4,6,7})	{1,2,4}
in_both({*ab*,*ac*,*ad*,*ae*},{*ae*,*ab*,*bc*})	{*ab*,*ae*}
in_one([1 2 3 4 5 10],[1 2 4 6 7])	{3,5,10,6,7}
in_both({*ab*,*ac*,*ad*,*ae*},[*ae* *ab* *bc*])	{*ab*,*ae*}
collatz_p(10)	{10,5,16,8,4,2,1}
collatz_p(100)	{100,50,25,76,38,19,58,29,88,44,22,11,34,17,52,26,13,40,20,10,5,16,8,4,2,1}
collatz_i(100)	["#" 100; "N" 25; "M" 100]
collatz_i(seq(*k*,*k*,1,100))	

"#"	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
"N"	0	1	7	2	5	8	16	3	19	6	14	9	9	17	17	4	12	20	20	7	7	15	15	10

I believe that the first screen is self-explaining. The results of *collatz_i* may serve for a pretty scatter plot.



The next screen shows the Vigenére-Code (see also DNL 39 from 2000). *div_sum*(*n*) gives the sum of all divisors of *n*. *next_p*(*n*) presents the next prime > *n* or < *n*. *is_pr*(*n*) needs some explanation.

vigenere("This is an endcoded message","Cryptography",10,1)	["VYGHBGGEEC" ; "KAQUCSFSYJ" ; "AVLOIFQWDL"]
vigenere("VYGHBGGEEC"&"KAQUCSFSYJ"&"AVLYXDNKTB","Cryptography",15,⁻1)	["THISISANENDCODE" ; "DMESSAGEAVMPVAN"]
vigenere("This is an endcoded message!","Cryptography",10,1)	"Error: Letters only for message"
{*next_p*(1000),*next_p*(⁻1000)}	{1009,997}
div_sum(100)	217
div_sum([100 200 ; 300 400])	[217 465 ; 868 961]
is_pr(2,seq(*x*,*x*,1,10))	{false,false,true,false,true,true,false,false,true,true}
is_pr(seq(*x*,*x*,1,12),13)	{false,true,false,false,false,true,true,false,false,false,true,false}
seq(mod(7^k,13),*k*,1,12)	{7,10,5,9,11,12,6,3,8,4,2,1}
seq(mod(6^k,13),*k*,1,12)	{6,10,8,9,2,12,7,3,5,4,11,1}
seq(mod(8^k,13),*k*,1,12)	{8,12,5,1,8,12,5,1,8,12,5,1}

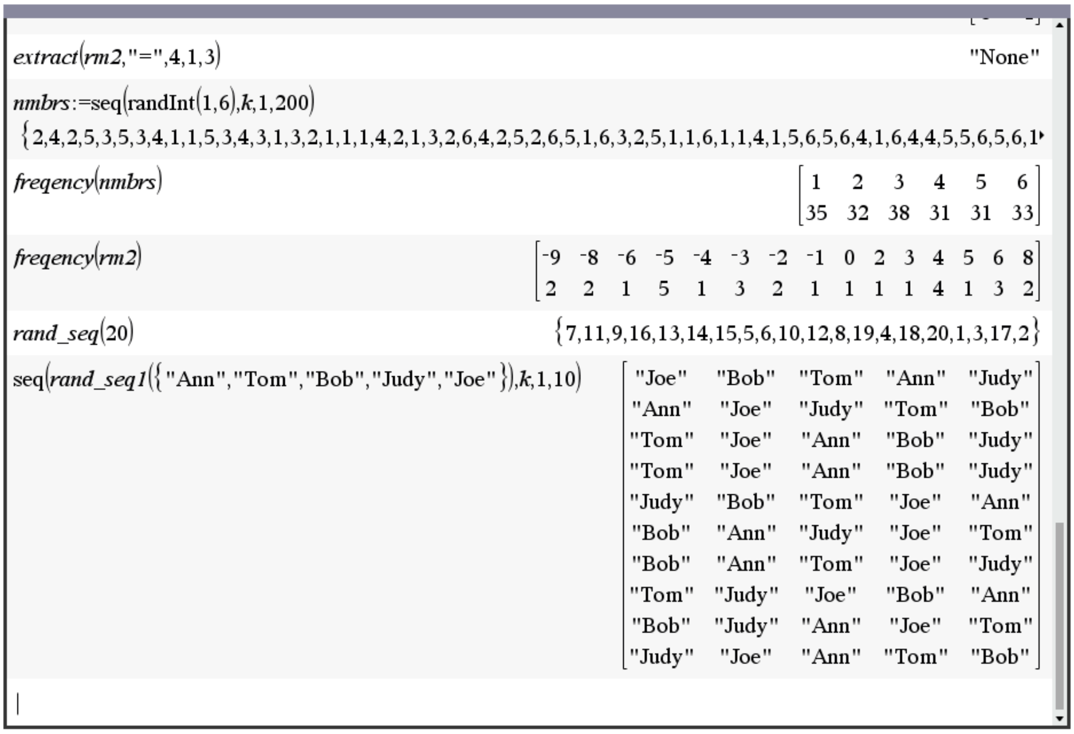
In number theory, a branch of mathematics, certain elements of prime residue class groups are called *primitive roots*. The defining property of a primitive root is that each element of the prime residue class group can be represented as one of its powers (Wikipedia).

The fourth expressions from below say us, that 2, 6, 7, and 11 are primitive roots of 13., i.e. all powers of 2, 6, 7 and 11 with exponents from 1 to 12 modulo 13 give all integers between 1 and 12 (compare results for bases 6 and 7 with this one for base 8, last line).

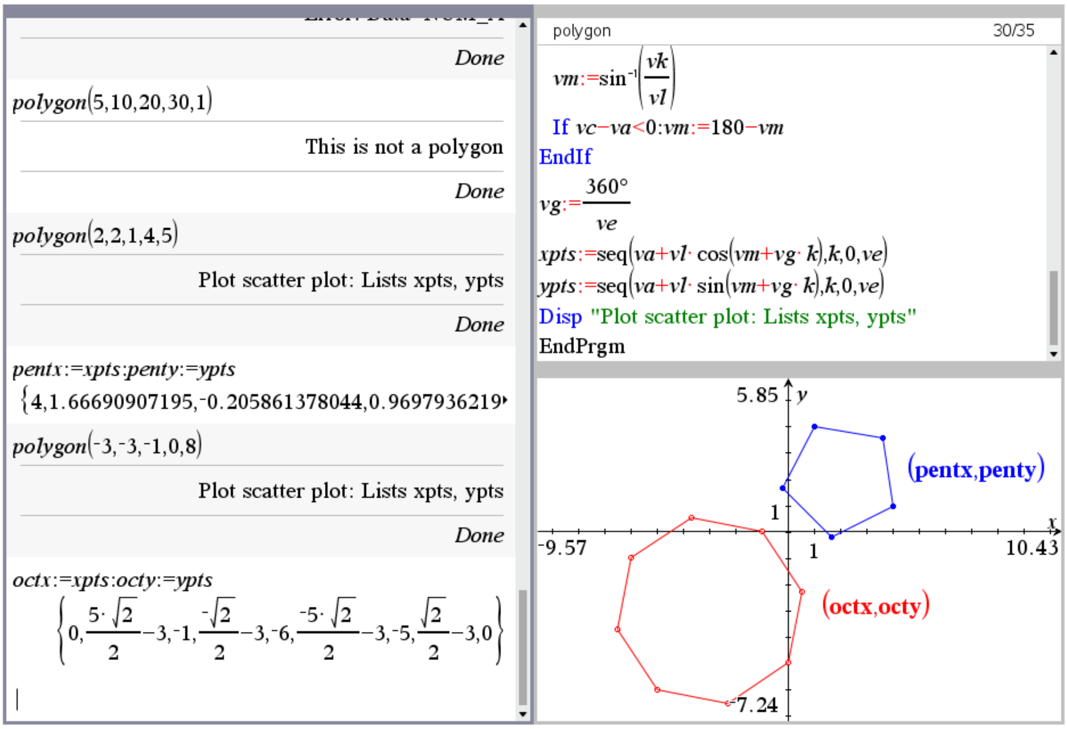
primes(1979,2021)	{1979,1987,1993,1997,1999,2003,2011,2017}
primes(100,50)	{97,89,83,79,73,71,67,61,59,53}
rm2:=randMat(6,5)	[4 ⁻3 ⁻1 5 ⁻5 ; ⁻6 ⁻5 ⁻3 ⁻2 ⁻9 ; 4 6 ⁻3 3 ⁻2 ; ⁻8 0 8 6 2 ; ⁻5 4 ⁻8 ⁻5 4 ; ⁻5 6 8 ⁻9 ⁻4]
extract(*rm2*,"<",0,3,1)	[⁻1 ⁻5 ; ⁻3 ⁻9 ; ⁻3 ⁻2 ; 8 2 ; ⁻8 4 ; 8 ⁻4]
extract(*rm2*,"=",4,1,3)	"None"
nmbrs:=seq(randInt(1,6),*k*,1,200)	{2,4,2,5,3,5,3,4,1,1,5,3,4,3,1,3,2,1,1,1,4,2,1,3,2,6,4,2,5,2,6,5,1,6,3,2,5,1,1,6,1,1,4,1,5,6,5,6,4,1,6,4,4,5,5,6,5,6,1▸
freqency(*nmbrs*)	[1 2 3 4 5 6 ; 35 32 38 31 31 33]

primes() and *freqency*() are obvious. The latter works not only for lists, but also for matrices. *extract*() is explained below.

extract(*rm2*,”<”,0,3,1) returns those columns of matrix rm2 with elements < 0 in row #3 after col #1. *extract*(*rm2*,”=”,4,1,3) asks for columns with a 4 in first line after column #3. Try to apply for rows!



rand_seq(*n*) or *rand_seq*(*list*) give a randomized sequence of numbers from 1 to *n* or of the elements given in *list*.



polygon(*xc*,*yc*,*xp*,*yp*,*n*) plots a regular *n*-edge with its center in (*xc*, *yc*) and one edge in (*xp*,*yp*). On the handhelds the polygons can be programmed for the Graph-screen directly using line-commands. This is not possible with TI-BASIC on Nspire. A scatter plot must do instead.

See another possibility performed with TI-Python (pages 45 and 46).

Michel sent his *Résume* and gave the permission for publication in our DNL. I had the idea to leave it in its original language – French. If I will find time enough during summer, I will start translating. I'll keep you informed and if so, then I'd send the English version as an attachment to one of my DUG-messages. Voilá, Josef

Michel Beaudin
michel.beaudin@etsmtl.ca
Version du 04.12.2020

Résumé 1
Résolution symbolique, résolution numérique d'équations et de systèmes d'équations
Rappels et compléments en équations différentielles

Nous pensons qu'une bonne façon de commencer le cours de maîtrise de mathématiques est de revenir sur la résolution d'équations/systèmes d'équations algébriques. En effet, cela est nécessaire à plusieurs endroits. On peut alors en profiter pour démontrer ou rappeler certains théorèmes, illustrer sur ordinateurs certaines méthodes et s'intéresser à la syntaxe des solveurs de différents systèmes symboliques. Mais on peut surtout en profiter pour introduire des concepts peu étudiés, par exemple parler de la formule de Cardan, de la fonction LambertW ou des systèmes polynomiaux tout en faisant des exercices sur ces sujets. Ces derniers points sont présentés de façon originale, puisant dans les ressources dont nous disposons maintenant pour enseigner les mathématiques.

Puisque que l'étude des systèmes d'équations différentielles et l'analyse de Fourier exigeront des rappels d'équations différentielles d'ordre un et deux, nous en faisons dans le présent résumé; et les itérations de Picard sont utilisées afin de renforcer le concept de solutions itératives. La transformée de Laplace est un outil fondamental et nous en rappelons les principales propriétés. Toutefois, ces « rappels » risquent d'être des compléments ou nouveautés pour certains étudiants car plusieurs exercices dépassent la théorie normalement vue dans un cours d'É.D. du premier cycle.

Voici les différentes sections de ce résumé.

1- Résolution symbolique d'équations et de systèmes d'équations

2- Méthode du point fixe et méthode de Newton

3- Équations différentielles : rappels et compléments

4- Transformée de Laplace : rappels et compléments

Liste d'exercices pour le résumé 1

1- Résolution symbolique d'équations et de systèmes d'équations

Cette section pourrait porter le titre de « la magie des systèmes symboliques » mais loin de nous la prétention d'expliquer comment fonctionnent ces systèmes. On veut plutôt guider l'étudiant dans l'utilisation de la fonction « solve » ou son équivalent. Le mieux est d'y aller avec certains exemples et en profiter pour faire des remarques, pertinentes nous l'espérons. Ces exemples sont importants pour l'étudiant qui suit Mat 805 et qui est peu ou pas familier avec les systèmes symboliques (logiciels de calcul symbolique, calculatrice symbolique). Il faut avoir en tête que le mode de calcul, par défaut, est le mode exact pour les systèmes sur ordinateur, alors que la calculatrice TI, en mode « auto », passe en numérique si le symbolique n'est pas possible ou utile.

1.1 Équation linéaire, équation quadratique

Considérons quelques équations « faciles » à résoudre. Disons pour commencer les équations polynomiales linéaire et quadratique :

$$a\,x+b=0,\ \ a\,x^2+b\,x+c=0\ \ (a,b,c\in\mathbb{R},\,a\neq 0).$$

Nous savons que les solutions sont respectivement $-\dfrac{b}{a}$ et $\dfrac{-b\pm\sqrt{b^2-4ac}}{2a}$. Rappelons que l'equation quadratique $a\,x^2+b\,x+c=0$ est transformée en $y^2-\dfrac{b^2-4ac}{4a^2}=0$ si l'on fait la substitution $x=y-\dfrac{b}{a}$: autrement dit, on a fait « disparaître » le terme en *x*, on a « complété le carré ». On en déduit ensuite la formule quadratique. *Maple* et *Derive* donnent ceci :

> $solve(a\,x+b=0,x);$

$$-\frac{b}{a}$$

> $solve(a\,x^2+b\,x+c=0,x);$

$$\frac{-b+\sqrt{-4\,a\,c+b^2}}{2\,a},\ -\frac{b+\sqrt{-4\,a\,c+b^2}}{2\,a}$$

#1: $\mathrm{SOLVE}(a\cdot x+b=0,\ x)$

#2: $x=-\dfrac{b}{a}$

#3: $\mathrm{SOLVE}(a\cdot x^2+b\cdot x+c=0,\ x)$

#4: $x=\dfrac{\sqrt{(b^2-4\cdot a\cdot c)}-b}{2\cdot a}\ \vee\ x=-\dfrac{\sqrt{(b^2-4\cdot a\cdot c)}+b}{2\cdot a}$

Figure 1.1

La calculatrice/logiciel symbolique de Texas Instruments (TI-Nspire CX CAS) :

solve$(a\cdot x+b=0,x)$	$x=\frac{-b}{a}$
solve$(a\cdot x^2+b\cdot x+c=0,x)$	$x=\frac{\sqrt{b^2-4\cdot a\cdot c}-b}{2\cdot a}$ or $x=\frac{-(\sqrt{b^2-4\cdot a\cdot c}+b)}{2\cdot a}$

Figure 1.2

1.2 Formule de De Moivre

Une autre équation « facile » à résoudre est la suivante : on fixe un nombre complexe z ainsi qu'un entier positif n et on résout l'équation $w^n = z$, équation dont les solutions sont appelées les « n racines n-ièmes du nombre z ». Évidemment pour z réel, des formules d'algèbre remarquable font le travail. Par exemple, pour résoudre $w^3 = -8$, il suffit d'utiliser la formule remarquable $a^3+b^3=(a+b)\left(a-ab+b^2\right)$ pour obtenir

$$w^3=-8 \Leftrightarrow w^3+8=0 \Leftrightarrow (w+2)\left(w^2-2w+4\right)=0 \Leftrightarrow w=-2 \text{ ou } w=1\pm i\sqrt{3}.$$

Un système symbolique est, en général, « orienté complexe » et, par conséquent, décide d'afficher pour « la » racine n-ième d'un nombre z la première solution de l'équation $w^n = z$ que l'on rencontre lorsqu'on part de la partie positive de l'axe réel et tourne dans le sens anti-horaire. Donc, « la » racine cubique de -8 sera $1+i\sqrt{3}$ et non pas -2 (donc toujours choisir la branche « principale » ou « rectangulaire » dans le logiciel Nspire dans les réglages de Format Réel ou Complexe). Donc, si z est quelconque, on ne simplifiera pas $(z^3)^{1/3}$ sauf si z est réel positif (ce qui donnera z) ou encore si z est réel négatif : on a alors

$$z<0 \Rightarrow \left(z^3\right)^{1/3}=-z\left(\frac{1}{2}+i\frac{\sqrt{3}}{2}\right).$$

De façon générale, la résolution de l'équation wn = z se fait par utilisation de la formule de De Moivre que nous rappelons ici. Soit $z = r\, e^{i\theta}$ un nombre complexe donné écrit sous forme polaire ($r \geq 0$ est son module et θ son argument : $-\pi < \theta \leq \pi$). Soit n est un entier positif et supposons qu'on cherche à résoudre l'équation $w^n = z$. On pose $w = R\, e^{i\varphi}$ de sorte que, par la formule de De Moivre,

$$w^n = R^n e^{ni\varphi}.$$

Mais alors

$$w^n = z \Leftrightarrow R^n e^{ni\varphi} = r\, e^{i\theta} \Leftrightarrow R=\sqrt[n]{r} \text{ et } \varphi=\frac{\theta+2k\pi}{n},\ k=0,1,2,\ldots,n-1.$$

Cela sera utile lorsque nous voudrons trouver les solutions d'une équation comme $2^x = x^6$. Nous pourrons nous concentrer sur l'équation $\alpha\, 2^{x/6} = x$ avec α parcourant l'ensemble des 6 racines 6- ièmes de l'unité. Réécrivant ensuite cette dernière équation sous la forme

$$\frac{x\ln(2)}{6}\, e^{-\frac{x\ln(2)}{6}} = -\alpha\frac{\ln(2)}{6},$$

il nous suffira de savoir résoudre une équation du type $w\, e^w = z$.

1.3 Exemple

Il semble bien que les systèmes symboliques utilisent la formule de De Moivre (ou quelque chose d'équivalent). Par exemple, les 6 racines sixièmes de l'unité sont données par les zéros de l'équation $z^6 - 1 = 0$: la prochaine figure montre aussi que pour avoir les solutions complexes avec Nspire, on lui indique en rajoutant un « c » devant « zeros » et pour afficher les zéros multiples, on doit utiliser la commande « cPolyRoots ».

$\text{zeros}(z^6-1,z)$	$\{-1,1\}$
$\text{cZeros}(z^6-1,z)$	$\left\{\frac{1}{2}+\frac{\sqrt{3}}{2}\cdot i,\frac{1}{2}-\frac{\sqrt{3}}{2}\cdot i,\frac{-1}{2}+\frac{\sqrt{3}}{2}\cdot i,\frac{-1}{2}-\frac{\sqrt{3}}{2}\cdot i,-1,1\right\}$
$\text{cZeros}((z^4-1)^2,z)$	$\{i,-i,-1,1\}$
$\text{cPolyRoots}((z^4-1)^2,z)$	$\{-1,-1,-i,-i,i,i,1,1\}$

Figure 1.3

1.4 Parties réelles, parties imaginaires et graphisme associé

Faisons quelques rappels sur les nombres complexes et profitons-en pour donner quelques informations. Posons i l'unité imaginaire. Rappelons que si $z = x + i\,y = r\,e^{i\theta}$, alors on écrit souvent ceci :

$$x = \text{Re}(z),\ \ y = \text{Im}(z)$$
$$r = |z| = \sqrt{x^2 + y^2}$$
$$\theta = \text{Arg}(z) = \frac{\pi}{2}\text{sign}(y) - \tan^{-1}\left(\frac{x}{y}\right)$$

Arg(z) est dans l'intervalle $]-\pi, \pi\,]$ et la formule donnée ci-haut couvre bien tous les cas (évidemment lorsque $y = 0$, alors z est réel et Arg(z) = 0 ou π dépendant du signe de z). Et si z_1, z_2 sont 2 nombres complexes, alors

$$z_1 = z_2 \Leftrightarrow \text{Re}(z_1) = \text{Re}(z_2) \wedge \text{Im}(z_1) = \text{Im}(z_2).$$

L'égalité de nombres complexes sera utilisée dans la recherche de solutions complexes à une équation à coefficients réels. Il suffira de remplacer la variable x par la variable complexe $x + iy$ et d'égaler les parties réelle et imaginaire de l'équation. Les solutions complexes de l'équation se trouveront là où les 2 courbes se croiseront. Cela sera difficile à réaliser avec Nspire CX CAS, le tracé implicite 2D étant confiné aux équations polynomiales pour l'instant. Cela a toujours été très facile avec *Derive*. Dans *Derive* (et aussi dans la TI), une variable non déclarée est supposée réelle: ainsi $\sqrt{x^2}$ se simplifie en |x|. Mais si l'on a déclaré x complexe (dans la TI, on peut écrire *x_* pour indiquer que la variable x est complexe), alors $\sqrt{x^2}$ reste tel quel lorsqu'on le simplifie.

1.4.1 Exemple Considérons la région du plan complexe définie par

$$0 < \mathrm{Im}\left(\frac{1}{z}\right) < 1.$$

Si l'on veut vérifier notre réponse en utilisant *Derive*, il suffit de faire cela : on indique au système que *z* est une variable complexe et trace le #2 (ou sa simplification #3) et ça donne la région ombrée plus bas dans la figure 1.4:

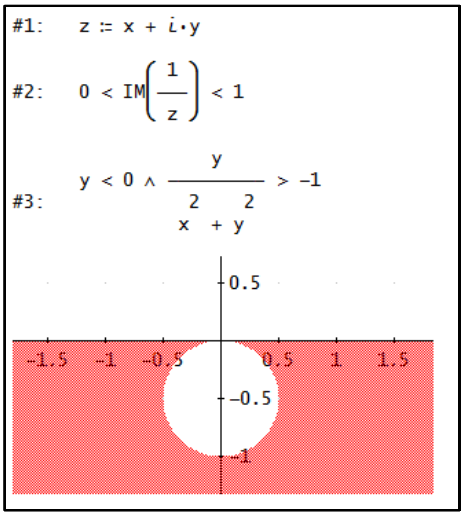


Figure 1.4

1.4.2 Exemple Prenons l'équation $x^3 + 3x + 1 = 0$ qui possède une seule racine réelle et deux racines complexes conjuguées. Cela est clair puisque la dérivée de $x^3 + 3x + 1$ est toujours positive : le polynôme changeant de signe entre −1 et 0, la racine réelle est donc située entre −1 et 0. La racine réelle ne peut être rationnelle en vertu d'un théorème d'algèbre bien connu. On va la trouver plus loin avec la formule de Cardan. Pour l'instant, montrons que l'utilisation des nombres complexes permet de « voir » les 3 racines. Remplaçons *x* par *x* + *iy* pur obtenir

$$(x+iy)^3 + 3(x+iy) + 1 = x^3 + 3x^2 iy + 3xi^2y^2 + i^3y^3 + 3x + 3iy + 1 =$$
$$= \left(x^3 - 3xy^2 + 3x + 1\right) + i\left(-y^3 + 3x^2y + 3y\right)$$

Il nous suffit de faire tracer, dans une même fenêtre 2D, les 2 courbes (implicites)
$x^3 - 3xy^2 + 3x + 1 = 0$ et $-y^3 + 3x^2y + 3y = 0$.

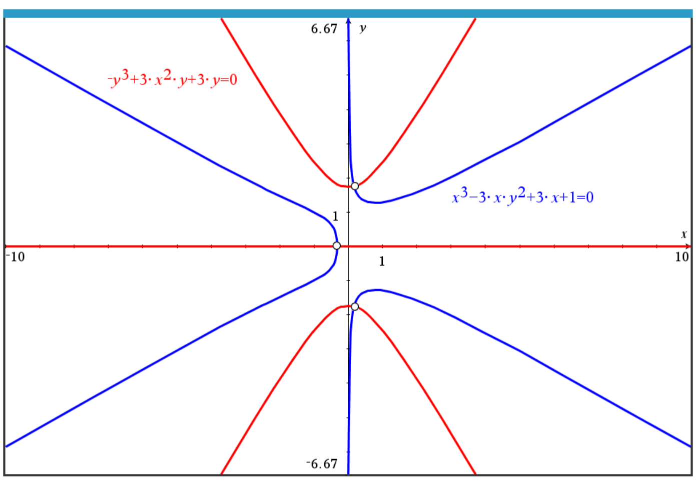


Figure 1.5

Ou encore de voir où la surface $z(x, y) = |(x+iy)^3 + 3(x+iy) + 1|$ touche au plan des *xy*:

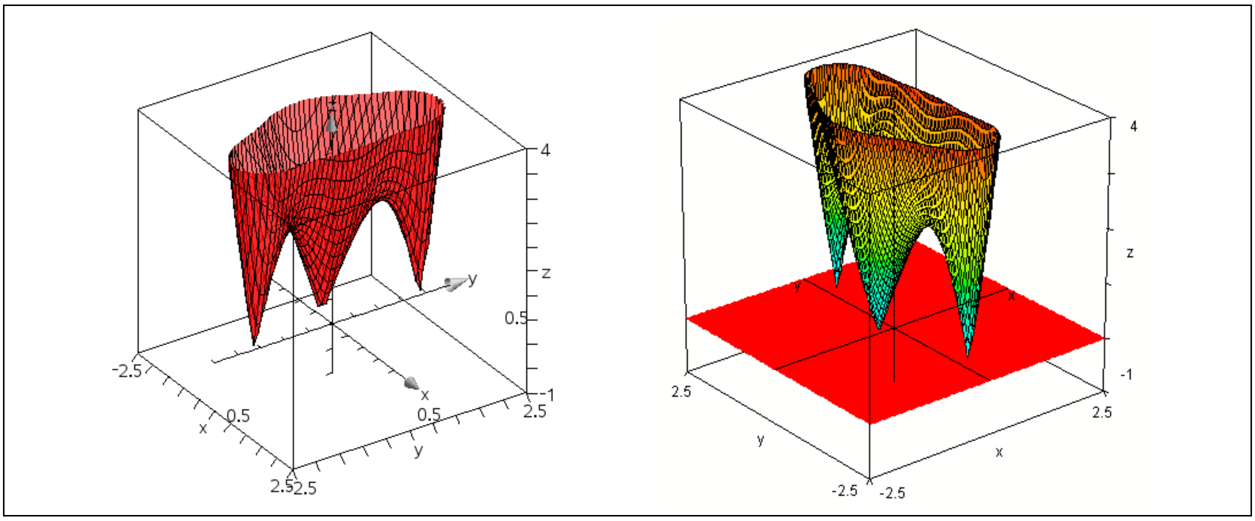


Figure 1.6

1.5 Formule de Cardan

Pour les équations polynomiales de degré 3 ou 4, des formules existent mais, dépendant du système, la réponse sera affichée sous des formes différentes. Dans certains cas, les formules de Cardan dont nous allons ici parler sont utilisées pour le degré 3.

Il y a des exemples où les 3 solutions sont réelles mais des radicaux imbriqués encombrent l'écran et la présence du nombre *i* $\left(\sqrt{-1}\right)$ laisse penser que 2 des 3 solutions sont complexes conjuguées alors que les 3 solutions sont réelles. Pour des raisons de commodité, la calculatrice symbolique passe en mode numérique plutôt que d'afficher des réponses encombrantes (à moins de forcer le mode « exact »). La structure « RootOf » de *Maple* joue un role important dans ce sens. De plus, les systèmes peuvent passer en arithmétique flottante avec des commandes du genre « fsolve », « nsolve ». Cela est souvent plus pratique.

Parlons de comment on résout l'équation polynomiale de degré trois. Il est facile de montrer que la substitution $x = y - \dfrac{b}{3a}$ transforme l'équation polynomiale du troisième degré

$$a x^3 + b x^2 + c x + d = 0$$

et l'équation $y^3 + 3py - 2q = 0$ où $p = \dfrac{c}{3a} - \dfrac{b^2}{9a^2}$ et $q = \dfrac{9abc - 27a^2 d - 2b^3}{54a^3}$. On peut donc se concentrer sur ne équation du type $y^3 + 3py - 2q = 0$ et évidemment $q \neq 0$. Trouvons cette solution réelle. Notez que d'écrire $y^3 + 3py - 2q = 0$ simplifiera la démonstration de l'importance du signe de $q^2 + p^3$ qui apparaîtra bientôt : de plus, la fonction $y^3 + 3py - 2q = 0$ aura deux points critiques lorsque $p < 0$: $\pm\sqrt{-p}$. Une belle animation de tout cela sera faite « live » au cours.

Voici une première façon de procéder : on pose $y = u - \dfrac{p}{u}$ et on aboutit à l'équation « quadratique déguisée » $u^6 - 2qu^3 - p^3$ dont on tire $u^3 = q + \sqrt{q^2 + p^3}$ (on a aussi $u^3 = q - \sqrt{q^2 + p^3}$ mais on peut montrer que les deux choix mènent au même résultat final). Cela permet de déduire la <u>formule de Cardan</u>:

$$y=\left(q+\sqrt{q^2+p^3}\right)^{1/3}-\frac{p}{\left(q+\sqrt{q^2+p^3}\right)^{1/3}}.$$

1.5.1 Remarque

Il est clair que si p est positif, il n'y a qu'une seule solution réelle (car la dérivée de $y^3+3py-2q$ est alors toujours positive). De plus $p^3 > 0$ et alors $\left(p^3\right)^{1/3} = p$. On peut donc réécrire la formule de Cardan comme suit <u>dans le cas où $p > 0$</u>:

$$\left(\sqrt{q^2+p^3}+q\right)^{1/3}-\frac{p}{\left(\sqrt{q^2+p^3}+q\right)^{1/3}}=\left(\sqrt{q^2+p^3}+q\right)^{1/3}-\frac{p\left(\sqrt{q^2+p^3}-q\right)^{1/3}}{\left(p^3\right)^{1/3}}=$$

$$=\left(\sqrt{q^2+p^3}+q\right)^{1/3}-\left(\sqrt{q^2+p^3}-q\right)^{1/3}$$

Notons que $p < 0$ n'implique pas nécessairement trois racines réelles. On montrera en classe que c'est le signe de $q^2 + p^3$ qui est déterminant et on comprend un peu pourquoi des nombres complexes ont été rendus nécessaires.

1.5.2 Exemple

Utilisons la formule de Cardan pour trouver l'unique solution réelle de l'équation $x^3 + 3x + 1 = 0$. On a $p = 1$ et $q = -\frac{1}{2}$ de telle sorte que $\sqrt{q^2+p^3}=\frac{\sqrt{5}}{2}$. Ainsi

$$x=\left(-\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}-\frac{1}{\left(-\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}}\approx -0.322185...$$

En utilisant la remarque précédente, on peut donc dire que l'unique solution réelle de l'équation $x^3 + 3x + 1 = 0$ est donnée par

$$x=\left(\frac{\sqrt{5}-1}{2}\right)^{1/3}-\left(\frac{\sqrt{5}+1}{2}\right)^{1/3}=\frac{\left(4\sqrt{5}-4\right)^{1/3}}{2}-\frac{\left(4\sqrt{5}+4\right)^{1/3}}{2}\approx -0.322185...$$

Le logiciel *Derive* ne faisait pas les choses à moitié:

```
                 3
#1:   SOLUTIONS(x  + 3·x + 1, x, Real)

                            1/3              1/3
                 ⎡ (4·√5 - 4)       (4·√5 + 4)   ⎤
#2:              ⎢ ───────────── - ───────────── ⎥
                 ⎣       2               2       ⎦
```

Figure 1.7

Voici une autre façon de procéder pour arriver à la formule de Cardan. L'idée est un peu comme celle de complétion du carré, donc ici une « complétion du cube » que nous appliquons à notre exemple. On veut résoudre $x^3 + 3x + 1 = 0$. On pose $x = u + v$ et alors

$$x^3+3x+1=(u+v)^3+3(u+v)+1=u^3+v^3+3uv(u+v)+3(u+v)+1=0.$$

On choisit de poser $u\,v = -1$. Cela donne $x^3+3x+1=u^3+v^3=0$. Résumons:

$$\begin{cases} u\,v=-1 \\ u^3+v^3=-1 \end{cases} \Rightarrow \begin{cases} u^3v^3=-1 \\ u^3+v^3=-1 \end{cases}$$

Or lorsque nous avons une équation quadratique de la forme $t^2 - (a + b)\,t + a\,b = 0$, nous avons que ses racines sont précisément a et b : cela nous mène donc à l'équation $t^2 + t - 1 = 0$ dons les solutions sont $t=-\frac{1}{2}\pm\frac{\sqrt{5}}{2}$. Donc $u^3=-\frac{1}{2}+\frac{\sqrt{5}}{2}$ et $v^3=-\frac{1}{2}-\frac{\sqrt{5}}{2}$. Pour $u^3=-\frac{1}{2}+\frac{\sqrt{5}}{2}$, la solution réelle est bien $\left(-\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}$ tandis que pour $v^3=-\frac{1}{2}+\frac{\sqrt{5}}{2}$ qui est négatif, la solution réelle est simplement $-\left(\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}$. On a encore $x=u+v=\left(-\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}-\left(\frac{1}{2}+\frac{\sqrt{5}}{2}\right)^{1/3}$.

1.5.3 Exemple

Dans le dernier exemple, changeons le signe du coefficient du terme linéaire et considérons plutôt l'équation $x^3 - 3x + 1 = 0$. Que fait *Maple*?

> $solve(x^3-3\,x+1=0,x);$

$$\frac{(-4+4\,I\sqrt{3})^{1/3}}{2}+\frac{2}{(-4+4\,I\sqrt{3})^{1/3}},\ -\frac{(-4+4\,I\sqrt{3})^{1/3}}{4}-\frac{1}{(-4+4\,I\sqrt{3})^{1/3}}+\frac{I\sqrt{3}\left(\frac{(-4+4\,I\sqrt{3})^{1/3}}{2}-\frac{2}{(-4+4\,I\sqrt{3})^{1/3}}\right)}{2},$$
$$-\frac{(-4+4\,I\sqrt{3})^{1/3}}{4}-\frac{1}{(-4+4\,I\sqrt{3})^{1/3}}-\frac{I\sqrt{3}\left(\frac{(-4+4\,I\sqrt{3})^{1/3}}{2}-\frac{2}{(-4+4\,I\sqrt{3})^{1/3}}\right)}{2}$$

$evalf(\%);$

$$1.532088886-1.\ 10^{-10}\,I,\ -1.879385241-1.732050808\ 10^{-10}\,I,\ 0.3472963549+1.732050808\ 10^{-10}\,I$$

> $fsolve(x^3-3\,x+1=0,x);$

$$-1.879385242,\ 0.3472963553,\ 1.532088886$$

Figure 1.8

Comme l'indique le « fsolve » de *Maple* dans la figure 1.8, l'équation possède bel et bien 3 racines réelles comme on peut d'ailleurs s'en convaincre avec un graphique ou par calcul différentiel. Soit dit en passant, notez qu'on trouve que $q^2 + p^3$ est négatif et on comprend alors pourquoi, comme on en parlait précédemment, les nombres complexes ont été inventés: pas pour résoudre l'équation du second degré mais celle du troisième degré! Les 3 solutions données par *Maple* sont conséquentes avec la formule de Cardan : en effet, ici on a les valeurs $p = -1$ et $q = -\frac{1}{2}$, de sorte que $u^3 = q + \sqrt{q^2 + p^3} = \frac{-1 + i\sqrt{3}}{2}$. Et la formule de De Moivre donne les 3 solutions pour u et ensuite on écrit $x = u - p/u$. Ces réponses peuvent être beaucoup plus élégantes… si l'on fait comme *Derive* qui semble utiliser une substitution trigonométrique puisque ses réponses sont les suivantes :

```
#1:   SOLVE(x^3 - 3·x + 1, x)

#2:            x = 2·COS(2·π/9) ∨ x = - 2·COS(π/9) ∨ x = 2·SIN(π/18)

#3:            x = 0.3472963553 ∨ x = 1.532088886 ∨ x = -1.879385241

#4:   NSOLVE(x^3 - 3·x + 1, x)

#5:            x = -1.879385241 ∨ x = 0.3472963553 ∨ x = 1.532088886
```

Figure 1.9

1.6 Équations du troisième degré dont les solutions sont toutes réelles

Montrons comment obtenir cela. On pose $x = a\sin(\theta)$ $(-\pi/2 \le \theta \le \pi/2)$ et substitue dans l'équation pour obtenir

$$a^3\sin^3\theta - 3a\sin\theta + 1 = 0.$$

Or, par identité trigonométrique, on a que $\sin^3\theta = \frac{3\sin\theta - \sin(3\theta)}{4}$ et ansi on obtient

$$a^3\left(\frac{3\sin\theta - \sin(3\theta)}{4}\right) - 3a\sin\theta + 1 = 0 \Leftrightarrow \left(\frac{3a^3}{4} - 3a\right)\sin\theta - \frac{a^3}{4}\sin(3\theta) + 1 = 0.$$

Choisir $a = 2$ rend l'équation facile à résoudre puisqu'alors $\sin(3\theta) = ½$.

Les solutions de $\sin(w) = a$ $(-1 \le a \le 1)$ sont $w = \arcsin(a) + 2k\pi$ et $w = \pi - \arcsin(a) + 2k\pi$.

Et puisque $\arcsin(1/2) = \pi/6$, les solutions d'une équation comme $\sin(w) = ½$ sont donc, à $2k\pi$ prés, $\pi/6$ et $5\pi/6$.

En faisant « rouler » k et devant rester entre $-\pi/2$ et $\pi/2$, on trouve $\theta = \pi/18$, $5\pi/18$ et $-5\pi/18$. Les 3 solutions des l'équation $x^3 - 3x + 1 = 0$ sont donc $2\sin\left(\frac{\pi}{18}\right)$, $2\sin\left(\frac{5\pi}{18}\right)$ et $-2\left(\frac{7\pi}{18}\right)$.

Mais $\sin(n\pi) = \cos((1/2 - n)\pi)$ et les réponses peuvent être remplacées par celles-ci :

$$2\sin\left(\frac{\pi}{18}\right), 2\cos\left(\frac{\pi}{9}\right) \text{ et } -2\cos\left(\frac{\pi}{9}\right).$$

Jetons un coup d'oeil à la calculatrice symbolique TI : notez que son « nsolve » ne donne ici qu'une réponse (c'est toujours le cas avec un solveur numérique). On le « guidera » pour obtenir les autres en donnant un point de départ ou une condition (voir écran plus loin). Notez aussi que Nspire CAS peut résoudre cette équation en mode exact, utilisant aussi des réponses sous forme trigonométrique … mais on est encore loin de l'élégance des réponses que *Derive* a données tantôt : pour cette raison, l'auteur de ces lignes a programmé des fonctions pour Nspire CAS :

$\text{nSolve}(x^3-3\cdot x+1=0,x)$	0.347296
$\text{nSolve}(x^3-3\cdot x+1=0,x=-2)$	-1.87939
$\text{nSolve}(x^3-3\cdot x+1=0,x)\vert x>0.4$	1.53209
$pol:=x^3-3\cdot x+1$	$x^3-3\cdot x+1$
$\text{zeros}(pol,x)$	$\left\{-\left(\cos\left(\frac{2\cdot\pi}{9}\right)+\sin\left(\frac{2\cdot\pi}{9}\right)\cdot\sqrt{3}\right),\frac{1}{\cos\left(\frac{2\cdot\pi}{9}\right)+\sin\left(\frac{2\cdot\pi}{9}\right)\cdot\sqrt{3}+1},\frac{\cos\left(\frac{2\cdot\pi}{9}\right)+\sin\left(\frac{2\cdot\pi}{9}\right)\cdot\sqrt{3}+1}{\cos\left(\frac{2\cdot\pi}{9}\right)+\sin\left(\frac{2\cdot\pi}{9}\right)\cdot\sqrt{3}}\right\}$
$kit\_ets\_mb\backslash compact\_cubic(pol,x)$	$\left\{2\cdot\sin\left(\frac{\pi}{18}\right),2\cdot\cos\left(\frac{2\cdot\pi}{9}\right),-2\cdot\cos\left(\frac{\pi}{9}\right)\right\}$

Figure 1.10

1.7 Exemples divers

Beaucoup de « trucs » sont implémentés dans les systèmes symboliques. Même s'il n'y a pas de formule pour une équation polynomiale de degré 5 ou plus, on peut quand même, dans certains cas, la résoudre. Considérons l'équation polynomiale suivante:$x^5 + x + 1 = 0$. Puisque la dérivée de la fonction continue $f(x) = x^5 + x + 1$ est toujours positive, alors il n'y a qu'une seule racine réelle, située entre -1 et 0 comme l'indique les valeurs de signe contraire des images de ces 2 points. La calculatrice symbolique TI distingue les solutions réelles des solutions complexes. De plus, il est souvent commode d'utiliser des fonctions semblables au « solve » mais dont la forme de la réponse est une liste ou matrice.

On voit à la figure 1.11 des saisies d'écrans provenant de *Derive*. Notamment 'utilisation de la commande "solve(eq, var, real)" et celle de "solutions(eq, var)". Des résultats numériques sont obtenus immédiatement en approximant "solve" plutôt qu'en le simplifiant ou en utilisant "nsolve" ou "nsolutions" plutôt que "solutions".

$$\text{SOLVE}(x^5 + x + 1 = 0,\ x) = \left[x = \frac{(100 - 12\cdot\sqrt{69})^{1/3}}{12} + \frac{(12\cdot\sqrt{69} + 100)^{1/3}}{12} + \frac{1}{3} + i\cdot\left(\frac{(108\cdot\sqrt{23} + 300\cdot\sqrt{3})^{1/3}}{12} - \frac{(300\cdot\sqrt{3} - 108\cdot\sqrt{23})^{1/3}}{12}\right) \vee x = \frac{(100 - 12\cdot\sqrt{69})^{1/3}}{12} + \frac{(12\cdot\sqrt{69} + 100)^{1/3}}{12} + \frac{1}{3} + i\cdot\left(\frac{(300\cdot\sqrt{3} - 108\cdot\sqrt{23})^{1/3}}{12} - \frac{(108\cdot\sqrt{23} + 300\cdot\sqrt{3})^{1/3}}{12}\right) \vee x = -\frac{1}{2} - \frac{\sqrt{3}\cdot i}{2} \vee x = -\frac{1}{2} + \frac{\sqrt{3}\cdot i}{2} \vee x = -\frac{(100 - 12\cdot\sqrt{69})^{1/3}}{6} - \frac{(12\cdot\sqrt{69} + 100)^{1/3}}{6} + \frac{1}{3}\right]$$

```
SOLVE(x^5 + x + 1 = 0, x, Real) = [x = - (100 - 12·√69)^(1/3)/6 - (12·√69 + 100)^(1/3)/6 + 1/3]

NSOLVE(x^5 + x + 1 = 0, x) = (x = 0.8774388331 - 0.7448617666·i ∨ x = 0.8774388331 + 0.7448617666·i ∨ x =
  -0.5 - 0.8660254037·i ∨ x = -0.5 + 0.8660254037·i ∨ x = -0.7548776662)

NSOLVE(x^5 + x + 1 = 0, x, Real) = (x = -0.7548776662)

NSOLUTIONS(x^5 + x + 1 = 0, x) = [-0.7548776662, -0.5 + 0.8660254037·i, -0.5 - 0.8660254037·i,
  0.8774388331 + 0.7448617666·i, 0.8774388331 - 0.7448617666·i]
```

Figure 1.11

Comment *Derive* a-t-il réussi à trouver de façon exacte la racine réelle de l'équation $x^5 + x + 1 = 0$? Difficile à dire puisque les systèmes symboliques sont très discrets sur les méthodes utilisées pour résoudre des équations. Ceux qui ont utilisé *Derive* savent toutefois que son intégrateur symbolique montrait les étapes de ses calculs et que cela se poursuit maintenant dans le système Rubi[*] d'Albert Rich. Tentons une explication sur la solution réelle de $x^5 + x + 1 = 0$. On a

$$\begin{aligned} x^5+x+1=0 &\Leftrightarrow x^5+x^4+x^3+x^2+x+1=x^4+x^3+x^2 \\ &\Leftrightarrow x^3(x^2+x+1)+(x^2+x+1)=x^2(x^2+x+1) \\ &\Leftrightarrow (x^3+1)\,(x^2+x+1)-x^2(x^2+x+1)=0 \\ &\Leftrightarrow (x^3-x^2+1)(x^2+x+1)=0 \end{aligned}$$

Puisque les racines du facteur quadratique sont complexes, la racine réelle de l'équation $x^5 + x + 1 = 0$ est donc la racine réelle de l'équation du troisième degré suivante :

$$pol(x)=x^3-x^2+1=0.$$

On se débarrasse du terme en x^2 en posant $x=y+\frac{1}{3}$ pour obtenir l'équation $y^3-\frac{y}{3}+\frac{25}{27}=0$ qui est donc de la forme $y^3+3py-2q=0$. Ici an a $p = -1/9$ et $q = -25/24$ de telle sorte que

$$q+\sqrt{q^2+p^3}=\frac{\sqrt{69}}{18}-\frac{25}{54}\approx -0.001484<0.$$

On sait (section 1.5) qu'on aura les 3 solutions de l'équation $y^3-\frac{y}{3}+\frac{25}{27}=0$ en résolvant l'équation $u^3=\frac{\sqrt{69}}{18}-\frac{25}{54}$ et ensuite $y=u-\frac{p}{u}$. Mais ici, ce dernier nombre est négatif et la branche principale étant toujours utilisée, la racine cubique de $\frac{\sqrt{69}}{18}-\frac{25}{54}$ sera un nombre complexe. On aura donc la racine réelle en multipliant par $e^{2\pi i/3}$: donc

$$u=e^{2\pi i/3}\left(\frac{\sqrt{69}}{18}-\frac{25}{54}\right)^{1/3}=\frac{\left(100-12\sqrt{69}\right)^{1/3}}{6} \quad \text{et} \quad y=u-\frac{p}{u}=-\frac{\left(100-12\sqrt{69}\right)^{1/3}}{6}-\frac{\left(100+12\sqrt{69}\right)^{1/3}}{6}.$$

Et puisque $x = y + 1/3$, on comprend d'où sortait la réponse de *Derive*! (Will be continued)

[*] https://rulebasedintegration.org/about.html

A Survey over Distributions – well known and not so well known 1

Bhuvanesh Bhatt & Josef Böhm

You must know that the original TI-92 and Voyage 200 had no single probability distribution implemented. But there was a Flash-Application "Statistics with List Editor" which contained many functions[*] which are now implemented in TI-Nspire. Nevertheless, I wanted to transfer Bhuvanesh Bhatt's functions because they are part of his great Mathtools-Library – and we can get a view inside the functions instead of using them as a black box only.

[*] https://education.ti.com/de-at/software/details/en/31FC737C43CF43B0ADA1CF67420C3AA8/89statisticswithlisteditor

Finally, the results of the implemented functions can serve as a reference for the "self-made" tools. Bhuvanesh put all distributions into two functions: PDF and CDF.

Working on this "translation" project I found out that there are much more distributions than treated in PDF and CDF. So, I added PDF2 and CDF2 (next DNL) with a second list of density and distribution functions.

Interestingly the important binomial distribution is missed. It is no problem to add it. I believe that it might be a good task for students to investigate some of the distributions which are not common in school mathematics in textbooks and/or internet (there are so many resources) and to find examples for their application. I did not include important properties like mean, variance, median etc. Students can be encouraged to find them out. Skewness, kurtosis, etc. could be discussed.

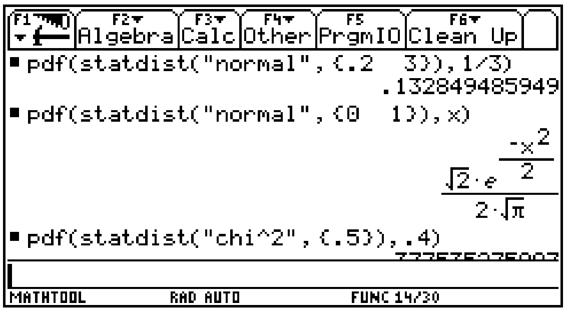
This is Bhuvanesh's description of his functions PDF and CDF for TI-89/ TI-92/ Voyage 200:

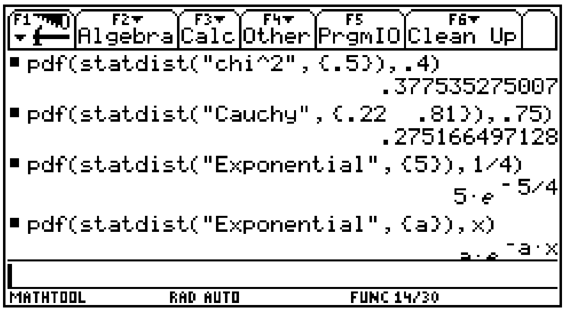
PDF(StatDist("DistributionName",parameters), x) returns the probability density function at *x* for the given distribution and parameters
Needs: Gamma

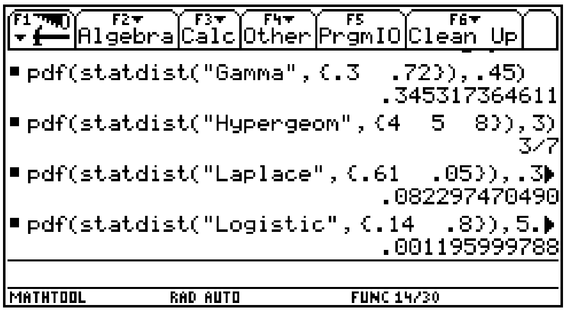
Examples:

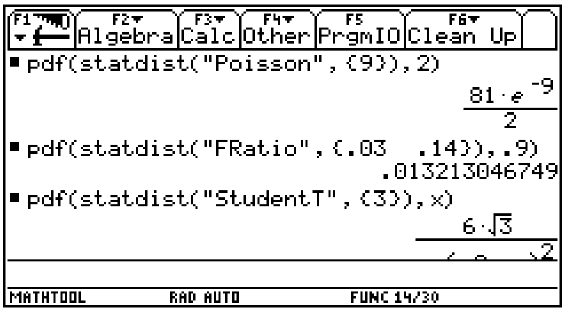
PDF(statdist("Normal",{0,1}),x) ⇒ √(2) e^(-x^2/2)/(2√(π)),
PDF(statdist("Normal",{0.2,3}),1/3) ⇒ 0.132849485949,
PDF(statdist("Exponential",{5}),1/4) ⇒ $5\cdot e^{-5/4}$,
PDF(statdist("Exponential",{a}),x) ⇒ $a\cdot e^{-a\cdot x}$,
PDF(statdist("Chi^2",{1/2}),0.4) ⇒ 0.377535275007,
PDF(statdist("Gamma",{0.3,0.72}),0.45) ⇒ 0.3453173646109662,
PDF(statdist("Hypergeometric",{4,5,8}),3) ⇒ 3/7,
PDF(statdist("Poisson",{9}),2) ⇒ $81\cdot e^{-9}/2$,
PDF(statdist("Cauchy",{0.22,0.81}),0.75) ⇒ 0.275166497128,
PDF(statdist("Laplace",{0.61,0.05}),0.37) ⇒ 0.08229747049,
PDF(statdist("Logistic",{0.14,0.8}),5.7) ⇒ 0.001195999788,
PDF(statdist("FRatio",{0.03,0.14}),0.9) ⇒ 0.013213046749, (F-Distribution)
PDF(statdist("StudentT",{0.66}),0.5) ⇒ 0.222461859264,
PDF(statdist("StudentT",{3}),x) ⇒ 6·√(3)/(π·(x^2+3)^2)

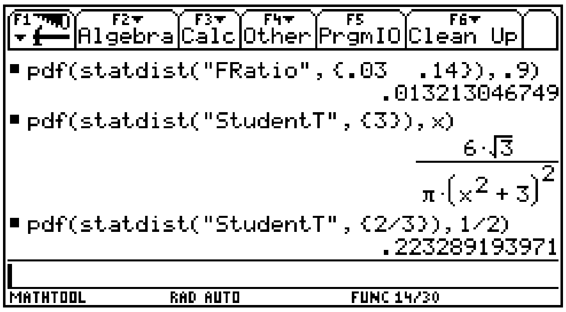
Here are some Voyage 200 screens showing applications of his function *pdf*:











CDF(StatDist("DistributionName", parameters), x) returns the distribution probability between -∞ and *x* for the given distribution and parameters

Needs: Erf, Erfc, Gamma, IncGamma, RegGamma, MatchQ

Examples:

CDF(statdist("Normal",{0,1}),0.2) ⇒ 0.579259709439,

CDF(statdist("Exponential",{0.83}),0.04) ⇒ 0.032654928773,

CDF(statdist("Chi^2",{0.3}),0.11) ⇒ 0.688767224365,

CDF(statdist("Gamma",{1/2,1/4}),2) ⇒ 0.999936657516,

CDF(statdist("Gamma",{1/2,1/4}),*x*) ⇒ 1-Γ(1/2,4*x*)/√(*π*),

CDF(statdist("Hypergeometric",{2,3,7}),1/3) ⇒ 2/7,

CDF(statdist("Poisson",{4/3}),9/4) ⇒ $29\cdot e^{-4/3}/9$

CDF(statdist("Cauchy",{1,2}),0.7) ⇒ 0.452606857723,

CDF(statdist("Cauchy",{1,2}),*x*) ⇒ $\tan^{-1}((x-1)/2)/\pi+1/2$,

CDF(statdist("Laplace",{0.96,0.32}),0.5) ⇒ 0.118760409548,

CDF(statdist("Logistic",{2.5,1}),4) ⇒ 0.817574476194,

CDF(statdist("FRatio",{0.03,0.14}),0.9) ⇒ 0.013213046749, (F-Distribution)

CDF(statdist("StudentT",{0.66}),0.5) ⇒ 0.222461859264,

CDF(statdist("StudentT",{3}),*x*) ⇒ $6\cdot\sqrt{3}/(\pi\cdot(x^2+3)^2)$

The red written distributions are not part of the original *cdf*-function. But the respective pdf-calls are contained in *pdf*(…). I added these distribution functions. In this article I will treat only from "Normal" to "Logistic". The remaining distributions will follow in one of the next DNLs. There are many other distributions worth to be treated, e.g. Pareto, Weibull, Beta, Triangle, Trapezium, Bernoulli, Geometric, Uniform, … and not to forget the Binomial distribution. I don't know why Bhuvanesh didn't include it in his function.

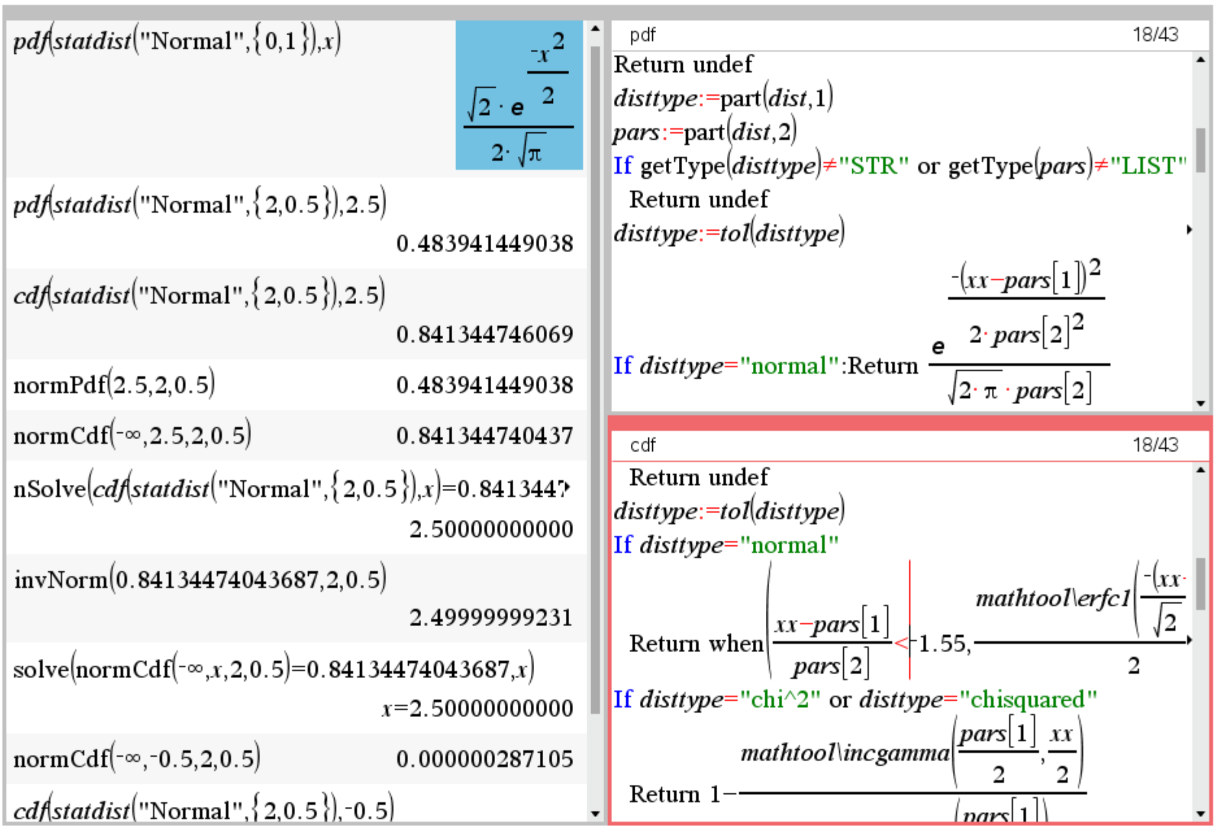
Bhuvanesh added another function, which also will be treated in one of the next DNLs:

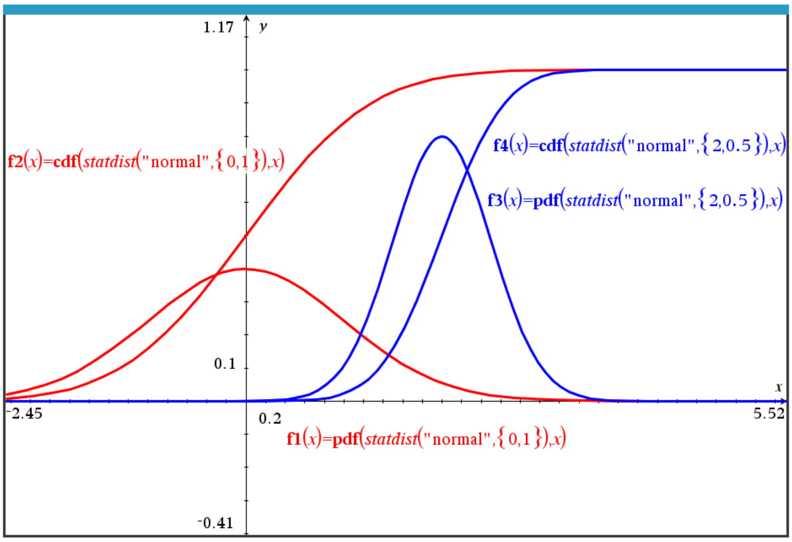
Random(StatDist("DistributionName",parameters)) returns a pseudorandom number using the given statistical distribution

Examples: RandSeed 0:Random(statdist("Bernoulli",{14})) ⇒ 1,
RandSeed 0:Random(statdist("Beta",{2.4,-0.11})) ⇒ 0.304489,
RandSeed 0:Random(statdist("Binomial",{4,2/3})) ⇒ 2,
RandSeed 0:Random(statdist("Cauchy",{-3/2,4})) ⇒ 20.8374,
RandSeed 0:Random(statdist("Chi^2",{1})) ⇒ 0.0603222,
…
…

As an introduction we will start with a very well-known continuous distribution It is the …

Normal distribution (& Erf, Erfi, Erfc and Gamma):





Plot of density and distribution function for two sets of parameters with TI-Nspire

DERIVE offers one function `NORMAL(x,m,s)` which is the distribution function. We know that its first derivative gives the density function:

```
NORMAL(3, 2, 0.5) = 0.9772498680

NSOLVE(NORMAL(x, 2, 0.5) = 0.977249868, x) = (x = 3)
```

$$\text{pdfnormal}(x, m, s) := \frac{d}{dx}\,\text{NORMAL}(x, m, s)$$

$$\text{pdfnormal}(3, 2, 0.5) = \frac{\sqrt{2}\cdot e^{-2}}{\sqrt{\pi}}$$

```
pdfnormal(3, 2, 0.5) = 0.1079819330
```

Inspecting the function code for the normal distribution and for the next distribution, which is χ^2, you will find Erfc1, Gamma and IncGamma used as auxiliary functions. There is also function *erf* in use which is *error function* defined by

$$\text{erf}\,(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,\text{d}t.$$

erfc(x) is the complimentary probability = 1 – erf(x). erfi1(x) is the complex error function

$$\text{erfi1}(x) = \frac{\text{erf1}(i\,x)}{i}.$$

Take care and distinguish:

erf1(2) = 0.995322,… erf1(2i) = 18.5648 i, erfi1(2) = 18.5648, erfi1(2i) = 0.995322 i

Comment: the original functions of Bhuvanesh were *erf*, *erfi* and *erfc*. Because they seem not to be completely correct, I tried to improve with the results: *erf1*, *erfi1* and *erfc1*.

$erf1(2)$	0.995322265019
$erf1(2\cdot i)$	$18.5648024146\cdot i$
$erfi1(2)$	18.5648024146
$erfi1(2\cdot i)$	$0.995322265019\cdot i$
$erf1(3+2\cdot i)$	$0.998963278901-0.000011546939\cdot i$
$erfi1(3+2\cdot i)$	$8.68731827074-20.8294614276\cdot i$
$erfc(2)$	0.004677734981
$erfc(3+2\cdot i)$	undef
$1-erf1(3+2\cdot i)$	$0.001036721099+0.000011546939\cdot i$
$erf1(2)$	0.995322265019
$erfc1(2)$	0.004677734981
$erfc1(3+2\cdot i)$	$0.001036721099+0.000011546939\cdot i$

erf1 1/2

Define LibPub **erf1**(zz)=
Func
© JB

$$\frac{2}{\sqrt{\pi}}\cdot\sum_{nn=0}^{400}\left(\frac{(-1)^{nn}\cdot zz^{2\cdot nn+1}}{nn!\cdot(2\cdot nn+1)}\right)$$

EndFunc

erfi1 1/2

Define LibPub **erfi1**(zz)=
Func
© JB

$$\frac{erf1(i\cdot zz)}{i}$$

EndFunc

WxMaxima serves as a very welcome reference:

```
(%i1) float(erf(2));
(%o1) 0.9953222650189527
(%i2) float(erf(2*%i));
(%o2) 18.56480241457555 %i
(%i3) float(erfi(2));
(%o3) -18.56480241457555
(%i4) float(erfi(2*%i));
(%o4) 0.9953222650189527 %i
(%i5) float(erf(3+2*%i));
(%o5) 0.9989632788568172-1.154672437929112 10^-5 %i
(%i6) float(erfi(3+2*%i));
(%o6) 8.687318271470174-20.82946142761456 %i
(%i7) float(erfc(3+2*%i));
(%o7) 1.154672437929112 10^-5 %i+0.00103672114318273
```

Derive:

ERF and ERFC are implemented, `erfi` must be defined:

$\mathrm{ERF}(2) = 0.9953222650$

$\mathrm{ERF}(2\cdot i) = 18.56480241\cdot i$

$\mathrm{ERF}(3 + 2\cdot i) = 0.9989632788 - 1.154672464\cdot 10^{-5}\cdot i$

$\mathrm{erfi}(z) := \dfrac{\mathrm{ERF}(i\cdot z)}{i}$

$\mathrm{erfi}(2) = 18.56480241$

$\mathrm{erfi}(2\cdot i) = 0.9953222650\cdot i$

$\mathrm{erfi}(3 + 2\cdot i) = 8.687318271 - 20.82946142\cdot i$

$\mathrm{ERFC}(2) = 0.004677734969$

$\mathrm{ERFC}(2\cdot i) = 1 - 18.56480241\cdot i$

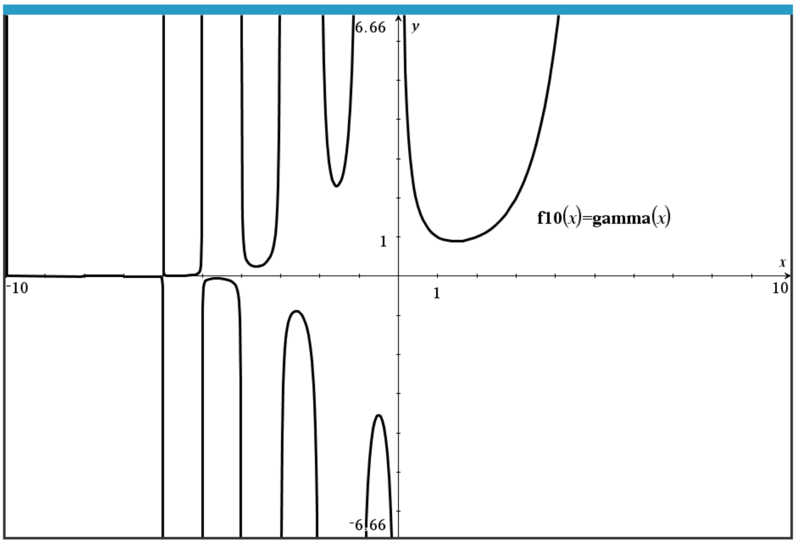
$\mathrm{ERFC}(3 + 2\cdot i) = 0.001036721144 + 1.154672464\cdot 10^{-5}\cdot i$

The Gamma-function is an extension of the factorial function to complex and non-integer numbers.

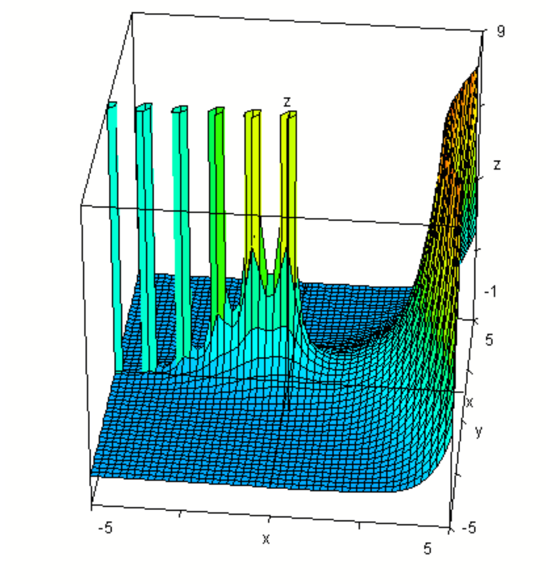
Gamma(z) = $\Gamma(z)$ is not defined for negative integers.

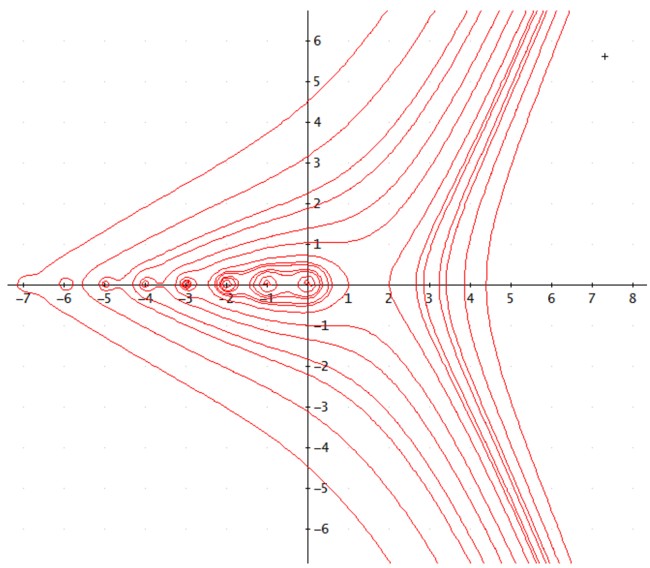
$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} \mathrm{d}x.$$

This function is one of the most important functions in mathematics which can be met in many fields of mathematics, especially in statistics. Many of the distributions presented in this survey need the Gamma function.



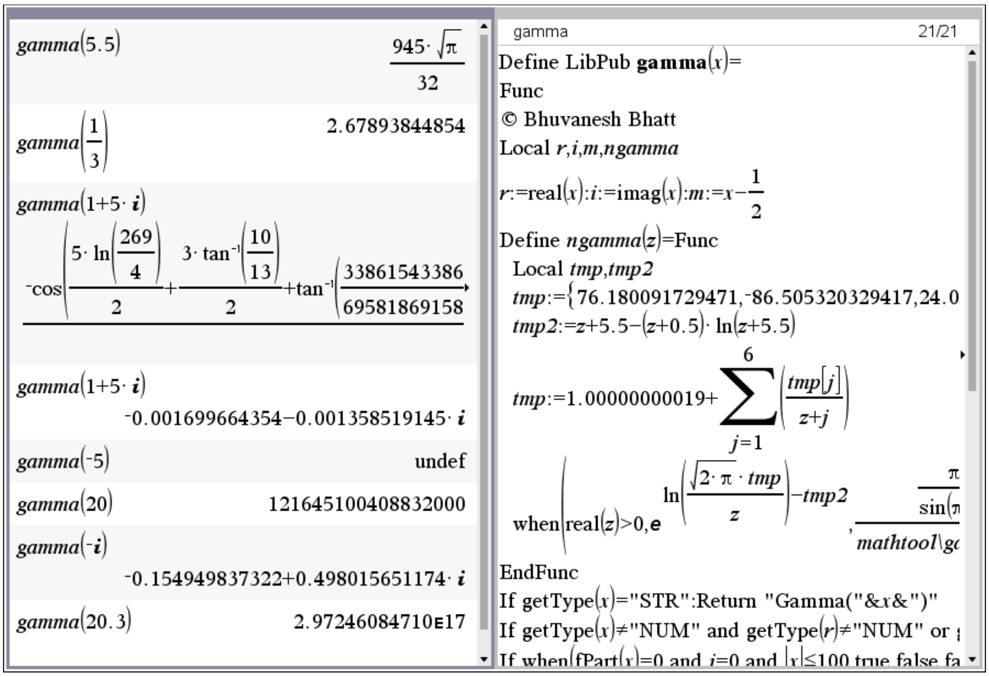
Graph of the Gamma function





Above a graph of the absolute value of this function with RE(z) and IM(z) as x- and y-axes together with some contour lines of this surface (*Derive*).

There is also an incomplete Gamma function and a regularized Gamma function. You will find many resources in the internet which inform about these functions. It is a bit confusing that their names are not the same in different resources.



Gamma function implemented on TI-NspireCAS

Derive:

$$\Gamma(5.5) = \frac{945\cdot\sqrt{\pi}}{32}$$

$$\Gamma\left(\frac{1}{3}\right) = 2.678938534$$

$$\Gamma(1 + 5\cdot i) = \left(\frac{1}{26} - \frac{5\cdot i}{26}\right)\cdot(1 + 5\cdot i)!$$

$$\Gamma(1 + 5\cdot i) = -0.001699664494 - 0.001358519417\cdot i$$

$$\Gamma(-5) = \pm\infty$$

$$\Gamma(20) = 121645100408832000$$

$$\Gamma(-i) = \left(-\frac{1}{2} + \frac{i}{2}\right)\cdot(1 - i)!$$

$$\Gamma(-i) = -0.1549498283 + 0.4980156681\cdot i$$

$$\Gamma(20.3) = \frac{3312042485029687115850686106542045067\cdot\left(\frac{3}{10}\right)!}{10000000000000000000}$$

$$\Gamma(20.3) = 2.972461075\cdot 10^{17}$$

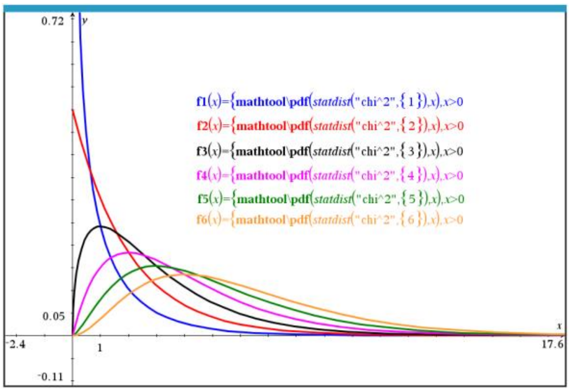
Gamma function is implemented in *Derive*

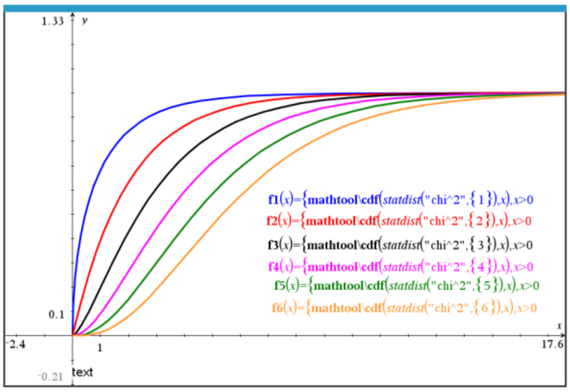
χ^2 distribution, Chi-square distribution

The χ^2-distribution with k degrees of freedom is the distribution of a sum of squares of k independent standard normal random variables. It is an often-used test distribution.

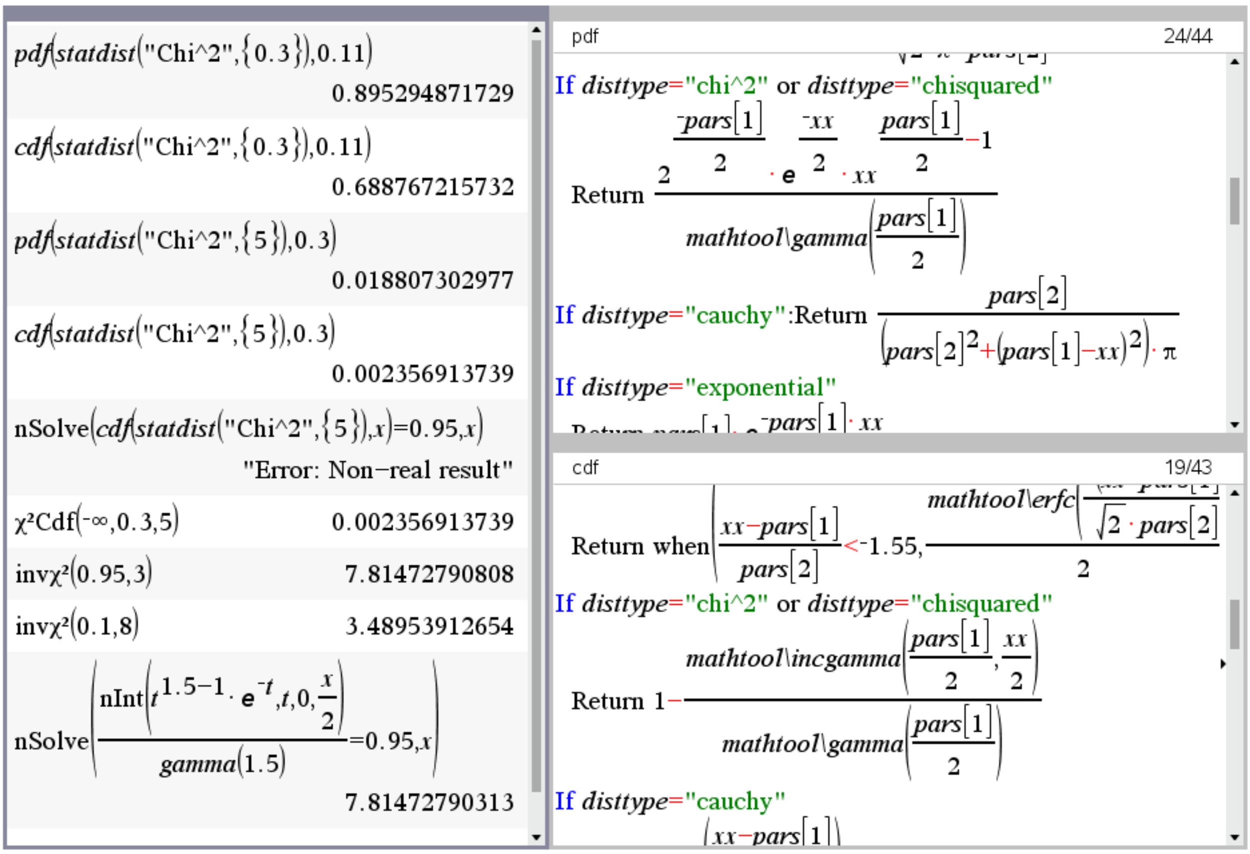
Density function: $f_k(X)=\dfrac{X^{k/2-1}\cdot e^{-x/2}}{2^{k/2}\cdot\Gamma(k/2)}$ $\quad X>0,\ 0$ else,

Distribution function: $F_k(x)=\dfrac{\int_0^{x/2} t^{k/2-1}\cdot e^{-t}\,dt}{\Gamma(k/2)}$ $\quad X>0,\ 0$ else





Density function (left) and distribution function (right) for various degrees of freedom (TI-Nspire)



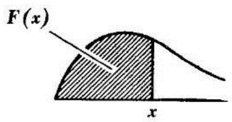
As you can see, this distribution function is implemented in TI-Nspire – pdf and cdf as well – and its inverse function, too. Bhuvanesh's function cannot be used to find the x-value for a given distribution value – but one can "nSolve" the respective equation using the definition given above.

In earlier times we had to use tables for all distributions:

402 ANHANG 5. ZAHLENTAFELN

6 **Chi-Quadrat-Verteilung**

Tafel 6. Werte von x zu gegebenen Werten der Verteilungsfunktion (60.3)



Beispiel: Bei 3 Freiheitsgraden ist $F = 0{,}99$ für $x = 11{,}34$.

$F(x)$	Anzahl der Freiheitsgrade 1	2	3	4	5	6	7	8	9	10
0,001	0,00	0,00	0,02	0,09	0,21	0,38	0,60	0,86	1,15	1,48
0,005	0,00	0,01	0,07	0,21	0,41	0,68	0,99	1,34	1,73	2,16
0,01	0,00	0,02	0,11	0,30	0,55	0,87	1,24	1,65	2,09	2,56
0,025	0,00	0,05	0,22	0,48	0,83	1,24	1,69	2,18	2,70	3,25
0,05	0,00	0,10	0,35	0,71	1,15	1,64	2,17	2,73	3,33	3,94
0,1	0,02	0,21	0,58	1,06	1,61	2,20	2,83	3,49	4,17	4,87
0,25	0,10	0,58	1,21	1,92	2,67	3,45	4,25	5,07	5,90	6,74
0,5	0,45	1,39	2,37	3,36	4,35	5,35	6,35	7,34	8,34	9,34
0,75	1,32	2,77	4,11	5,39	6,63	7,84	9,04	10,22	11,39	12,55
0,9	2,71	4,61	6,25	7,78	9,24	10,64	12,02	13,36	14,68	15,99
0,95	3,84	5,99	7,81	9,49	11,07	12,59	14,07	15,51	16,92	18,31
0,975	5,02	7,38	9,35	11,14	12,83	14,45	16,01	17,53	19,02	20,48
0,99	6,63	9,21	11,34	13,28	15,09	16,81	18,48	20,09	21,67	23,21
0,995	7,88	10,60	12,84	14,86	16,75	18,55	20,28	21,96	23,59	25,19
0,999	10,83	13,82	16,27	18,47	20,52	22,46	24,32	26,13	27,88	29,59

This is the χ^2-distribution table from *Erwin Kreyszig, Statistische Methoden und ihre Anwendungen*, one of the earlier standard text books for statistics. Try to find the values 7.81 and 3.49 from above in the table.

Let's look how *Derive* can perform?

#1: $$\text{CHI\_SQUARE}(0.3,\ 5) = \frac{\sqrt{5}\cdot e^{-3/20}\cdot\left(10\cdot\sqrt{5}\cdot\sqrt{\pi}\cdot e^{3/20}\cdot \text{ERF}\left(\frac{\sqrt{15}}{10}\right) - 11\cdot\sqrt{3}\right)}{50\cdot\sqrt{\pi}}$$

#2: `CHI_SQUARE(0.3, 5) = 0.002356913738`

#3: $$\text{pdf\_x\_square}(x,\ k) := \frac{d}{dx}\ \text{CHI\_SQUARE}(x,\ k)$$

#4: `pdf_x_square(0.3, 5) = 0.01880730297`

#5: $$\text{pdf\_x\_square}\left(0.4,\ \frac{1}{2}\right) = 0.3775352750$$

#6: `CHI_SQUARE(0.11, 0.3) = 0.6887672242`

#7: `NSOLVE(CHI_SQUARE(x, 3) = 0.95, x) = (x = 7.814728021)`

#8: `NSOLVE(CHI_SQUARE(x, 8) = 0.1, x) = (x = -2.026205472)`

#9: `NSOLVE(CHI_SQUARE(x, 8) = 0.1, x, 0, 10) = (x = 3.489539176)`

The density function is not implemented but can easily be defined as derivative of the distribution function `CHI_SQUARE(x,k)`.

Expression #8 gives a negative solution but we can force the solver to find the positive one!

Example:

I observe a digital clock in irregular time intervals and note the seconds rounded to the next ten. After doing this 100 times, I get the following table:

Seconds	0	10	20	30	40	50
Frequency	16	19	18	17	17	12

The observed frequencies should be close together. Can we confirm this assumption (α = 5%)?

The theoretical frequencies of numbers 0, 10, … are 100/6. We calculate:

$\chi^2 = \sum_{i=1}^{6} \frac{(o_i - e_i)}{e_i}$ with o_i = observed frequency and e_i = expected frequency of outcome i.

$$\chi^2 = \frac{\left(16-\frac{100}{6}\right)^2 + \left(19-\frac{100}{6}\right)^2 + \ldots + \left(13-\frac{100}{6}\right)^2}{\frac{100}{6}} = \frac{32}{25} = 1.28$$

6 variables, i.e. 5 degrees of freedom:

```
nSolve(cdf(statdist("Chi^2",{5}),x)=0.95,x)
                                  "Error: Non−real result"
invχ²(0.95,5)                                11.0704976927
nSolve( nInt(t^(2.5−1)·e^(−t),t,0,x/2) / gamma(2.5) =0.95,x)
                                             11.0704976935
```

1.28 < 11.07 ⇒ we can confirm that the outcomes are approximately equally distributed.

Exponential distribution

This distribution has many applications from everyday life., e.g. time between telephone calls, machinery service life, time until a radioactive particle decays, time between customers at a counter, and much more. (https://e.wikipedia.org/wiki/Exponential_distribution)

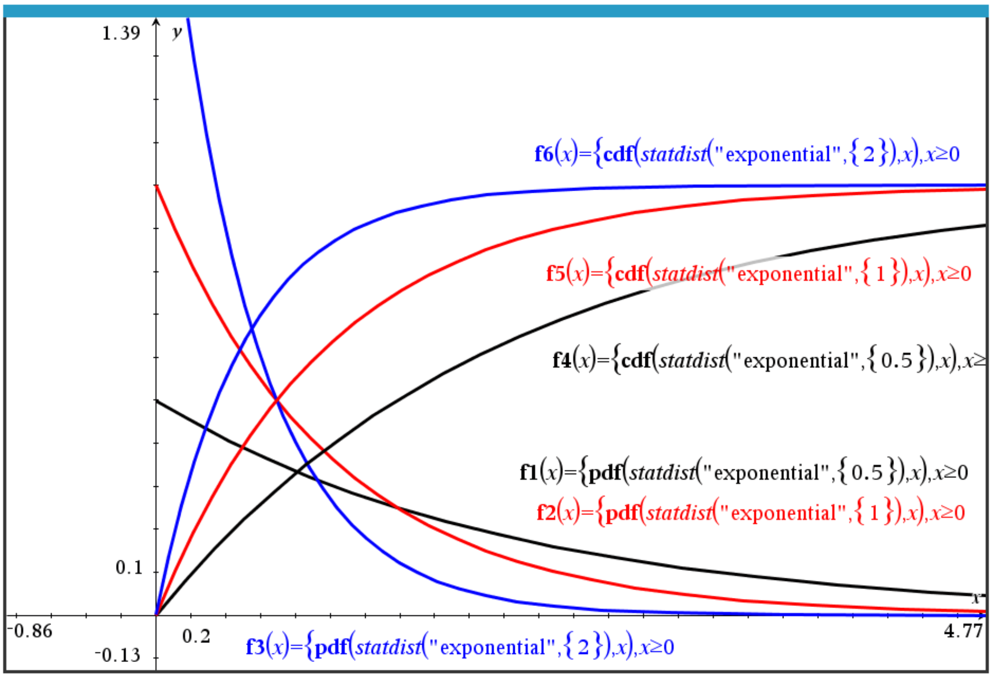
Density function: $f_\lambda(X) = \lambda \cdot e^{-\lambda X}$, $X \geq 0$, 0 else, Distribution function: $F_\lambda(x) = 1 - e^{-\lambda x}$, $x \geq 0$, 0 else

Two examples:

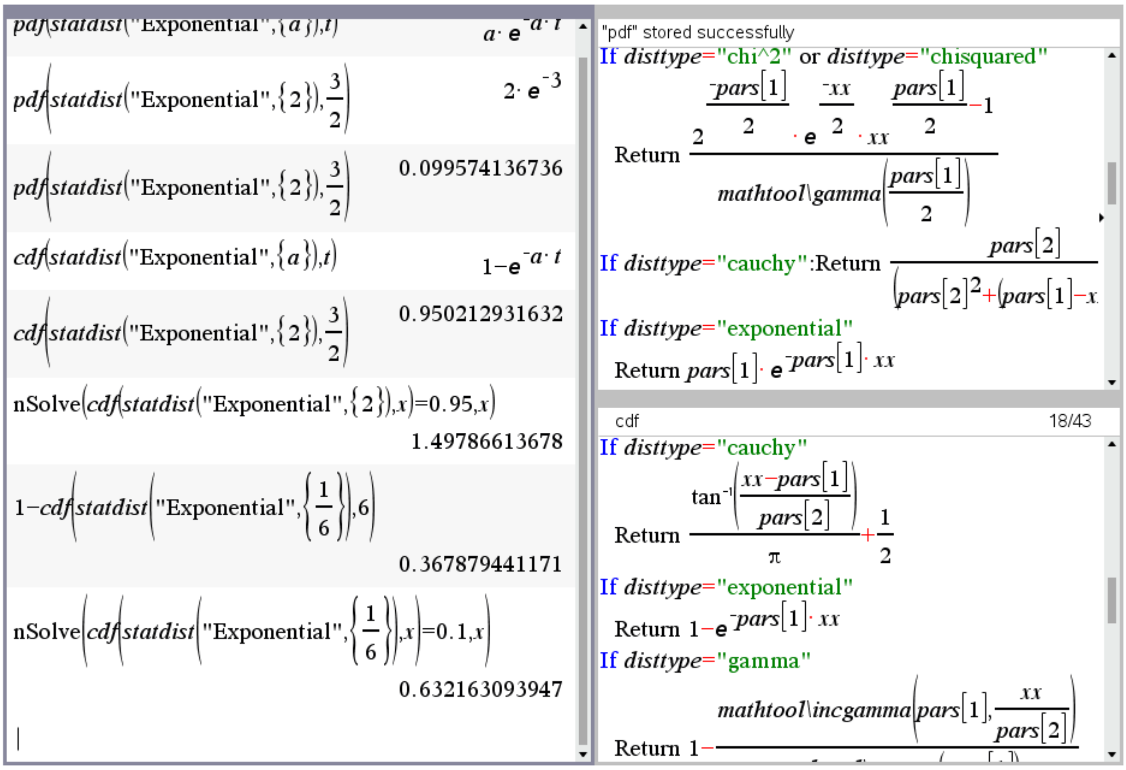
Lifetime of a machine follows an exponential distribution with expected value of lifetime 6 yrs.
So, λ = 1/6. What is the probability that a machine will survive its expected lifetime?

Which warranty period should be given, if only 10% machine breakdowns should be expected with this period?

The answers are given in the last two expressions in the screen presented below:
The probability is ~ 37% and the warranty period should be between 7 and 8 months.



Some pdfs and cdfs of the exponential distribution



Exponential distribution on TI-Nspire. Solutions of the life time problem is given in the last two expressions.

It is no problem to define pdf and cdf in *Derive*.

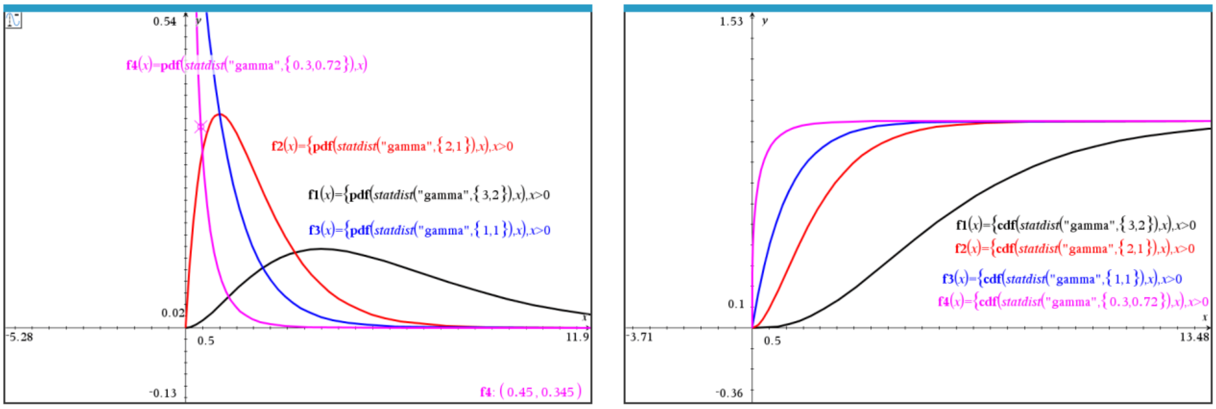
Gamma distribution

The Gamma distribution is a generalization of the exponential distribution. In queuing theory, it serves to describe repair and service times, in actuarial mathematics it is a model for small and medium damages.

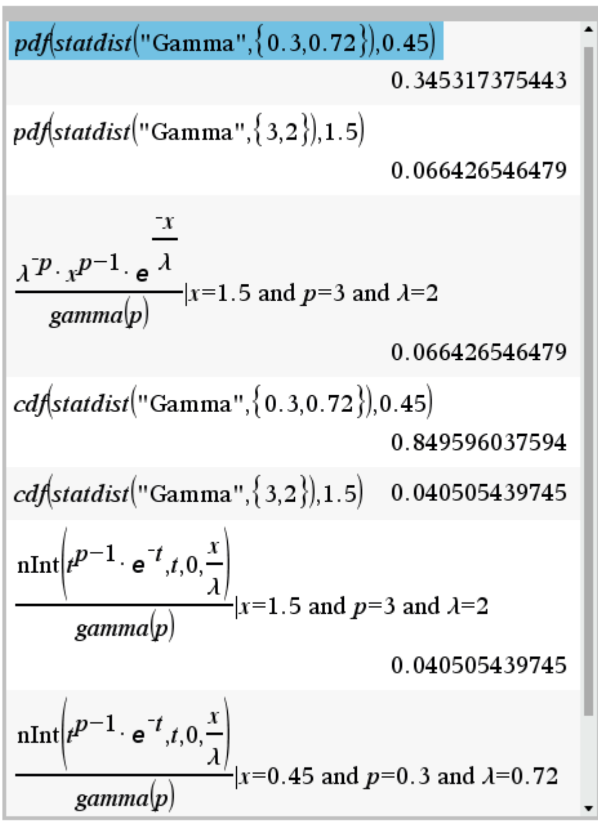
https://www.fm.mathematik.uni-muenchen.de/teaching/teaching_winter_term_2017_18/lectures_17_18/schadenvers/svm-teil1.pdf

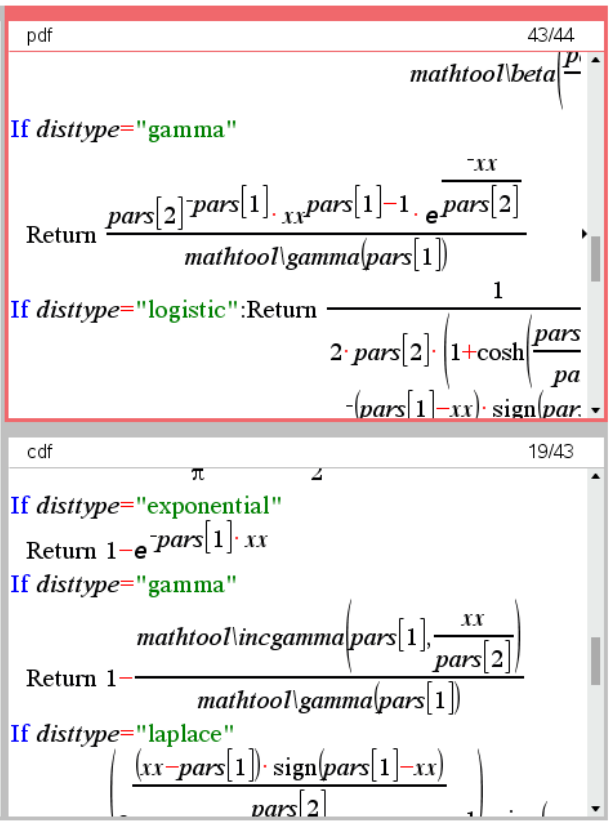
Density function: $f_{p,\lambda}(X)=\dfrac{\lambda^{-p}\cdot X^{p-1}\cdot e^{-X/\lambda}}{\Gamma(p)}$, $X>0$, 0 else,

Distribution function: $F_{p,\lambda}(x)=\dfrac{\int_0^{x/\lambda} t^{p-1}\cdot e^{-t}\mathrm{d}t}{\Gamma(p)}$, $x>0$, 0 else



Graphs of density and distribution function for various choices for p and λ (TI-Nspire)





Derive:

$$\text{pdf\_gam}(p, \lambda, x) := \frac{\lambda^{-p} \cdot x^{p-1} \cdot e^{-x/\lambda}}{\Gamma(p)}$$

$$\text{pdf\_gam}(3, 2, 1.5) = \frac{9 \cdot e^{-3/4}}{64}$$

$$\text{pdf\_gam}(3, 2, 1.5) = 0.06642654647$$

$$\text{pdf\_gam}(0.3, 0.72, 0.45) = \frac{250^{1/10} \cdot e^{-5/8}}{3 \cdot \left(\frac{3}{10}\right)!}$$

$$\text{cdf\_gam}(p, \lambda, x) := \text{INCOMPLETE\_GAMMA}\left(p, \frac{x}{\lambda}\right)$$

$$\text{cdf\_gam}(3, 2, 1.5) = \frac{e^{-3/4} \cdot (32 \cdot e^{3/4} - 65)}{32}$$

$$\text{cdf\_gam}(3, 2, 1.5) = 0.04050543974$$

$$\text{cdf\_gam}(0.3, 0.72, 0.45) = 0.8495960409$$

NSOLVE(cdf_gam(3, 2, x) = 0.5, x)

x = 5.348120629

Solving the inverse problem (last line) works on TI-Nspire, too.

https://en.wikipedia.org/wiki/Gamma_distribution

https://de.wikipedia.org/wiki/Gammaverteilung

https://www.wolframalpha.com/input/?i=Gamma+distribution&x=0&y=0

Hypergeometric distribution

This is the first discrete probability distribution in this survey. It describes the probability getting x objects with a certain property in sample of size n from a population of size N that contains M objects with that property.

Density function: $P(X=k) = \dfrac{\binom{M}{k} \cdot \binom{N-M}{n-k}}{\binom{N}{n}}$ with $k \in \{\max(0, n+M-N), \ldots, \min(n, M)\}$,

Distribution function: $P(X \le k) = \sum_{i=0}^{k} \dfrac{\binom{M}{i} \cdot \binom{N-M}{n-i}}{\binom{N}{n}}$

Example:

Twenty compact discs are chosen at random from special offer of 200 CDs of which 25 are defective.

a) What is the probability that three CDs among the chosen ones are defective?

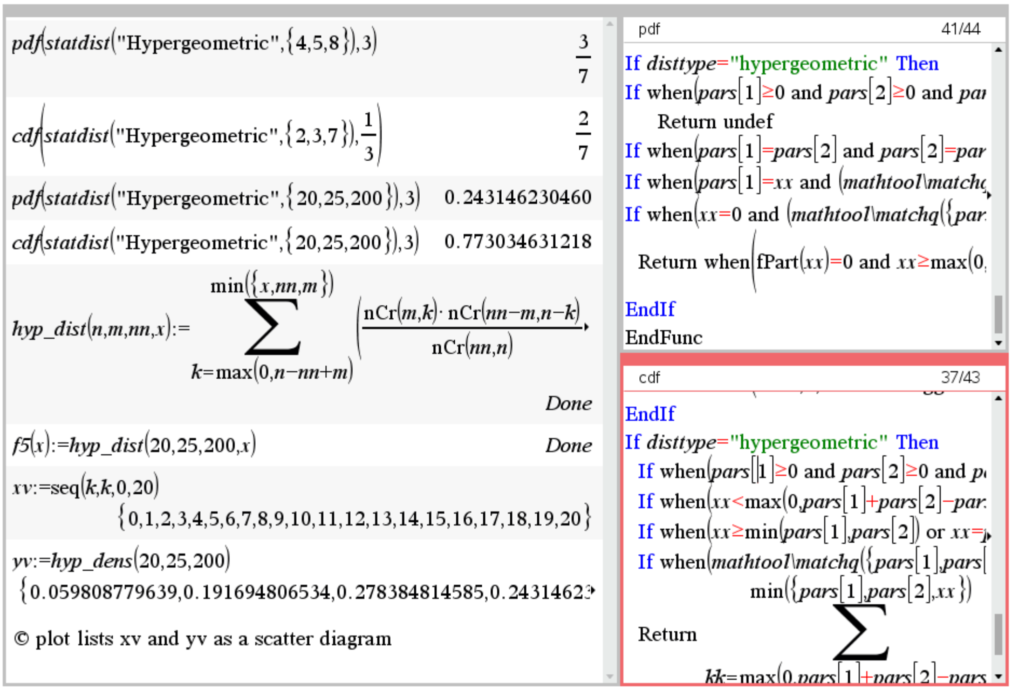
b) What is the probability that at most three CDs are defective?

Comment: This is a very typical problem from school mathematics.

a) $P(X=3) = \dfrac{\binom{25}{3} \cdot \binom{175}{17}}{\binom{300}{20}} \approx 0.2431$

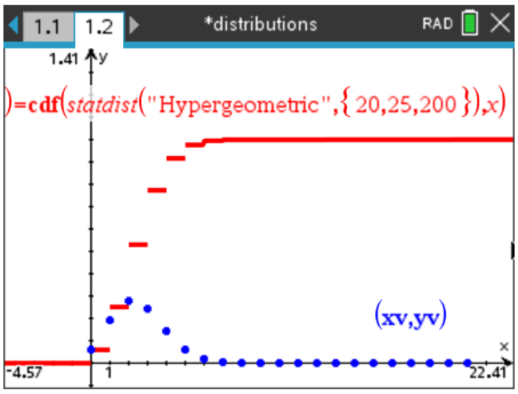
b) $$P(X \le 3) = \frac{\binom{25}{0}\cdot\binom{175}{20}+\binom{25}{1}\cdot\binom{175}{19}+\binom{25}{2}\cdot\binom{175}{18}+\binom{25}{3}\cdot\binom{175}{17}}{\binom{200}{20}} \approx 0.7730$$

Both calculations were performed using the TI-Voyage200 calculator. And now with the function:



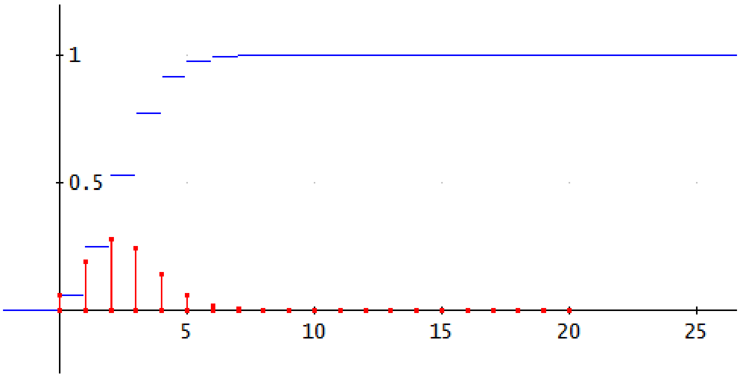
Two lists (*xv* and *yv*) are needed for plotting the density function and they are used for defining a scatter plot.

pdf and cdf presented in Nspire's Graphs-application.



This distribution is implemented in *Derive*. A short VECTOR-construction gives a nice plot of the density function:

```
HYPERGEOMETRIC_DENSITY(3, 20, 25, 200)

0.2431462304

HYPERGEOMETRIC_DISTRIBUTION(3, 20, 25, 200)

0.7730346312

HYPERGEOMETRIC_DISTRIBUTION(x, 20, 25, 200)

VECTOR([[k, 0], [k, HYPERGEOMETRIC_DENSITY(k, 20, 25, 200)]], k, 0, 20)
```



Hypergeometric pdf and cdf with *Derive*

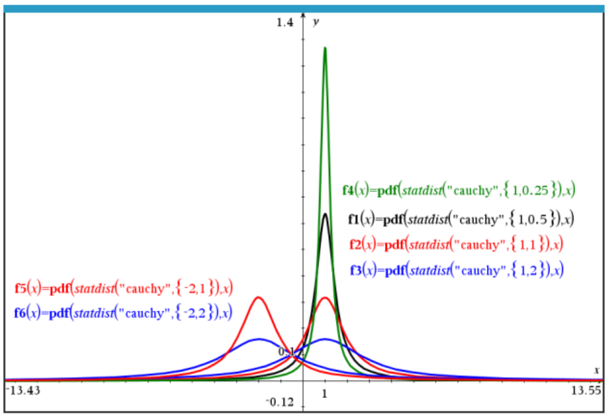
Cauchy distribution

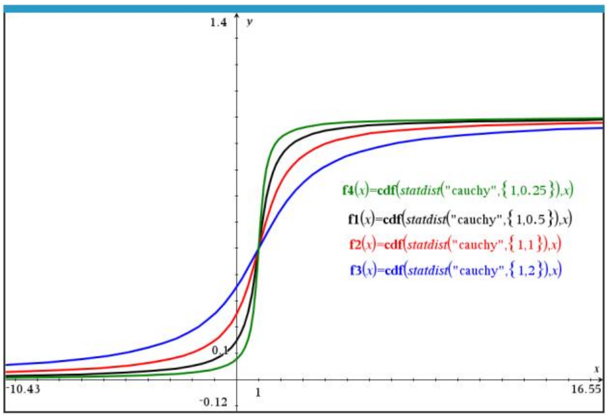
The Cauchy distribution has been used in many applications such as mechanical and electrical theory, physical anthropology, measurement problems, risk and financial analysis. In hydrology the Cauchy distribution is applied to extreme events such as annual maximum one-day rainfalls and river discharges.

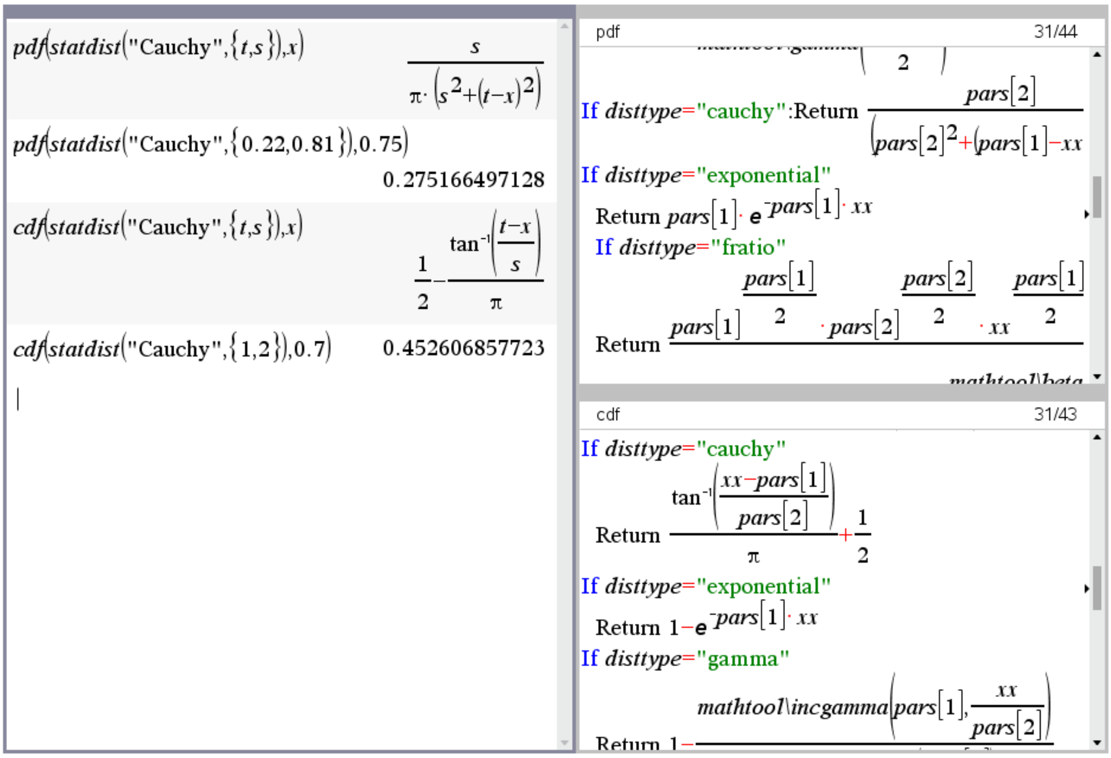
As an additional distribution to model fat tails in computational finance, Cauchy distributions can be used to model VAR (value at risk) producing a much larger probability of extreme risk than Gaussian Distribution (Wikipedia)

Density function: $f_{s,t}(X) = \frac{1}{\pi} \cdot \frac{s}{s^2 + (x-t)^2}, -\infty < X < \infty, \; s > 0, t \in \mathbb{R}$

Distribution function: $Fs,t(x) = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{x-t}{s}\right)$







Cauchy distribution with *Derive*:

$$\#1:\quad \text{pdf\_cauchy}(t, s, x) := \frac{1}{\pi}\cdot\frac{s}{s^2 + (x - t)^2}$$

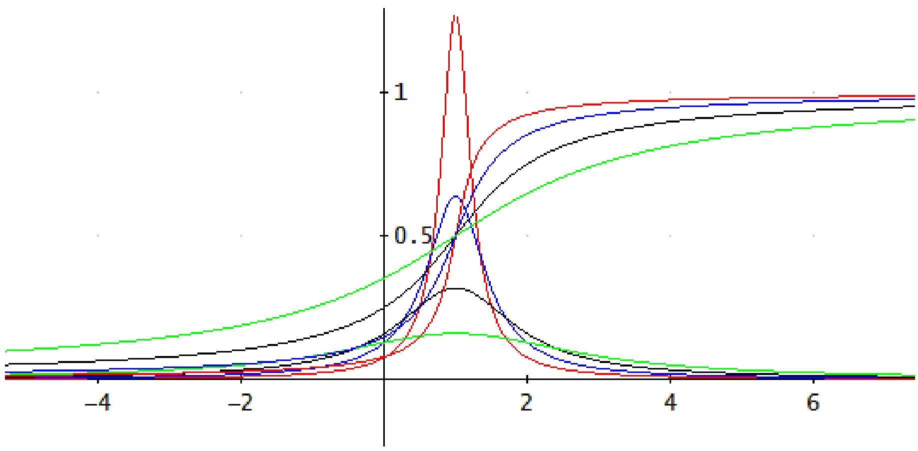
$$\#2:\quad \text{cdf\_cauchy}(t, s, x) := \frac{1}{2} + \frac{1}{\pi}\cdot \text{ATAN}\left(\frac{x - t}{s}\right)$$

#3: pdf_cauchy(0.22, 0.81, 0.75) = 0.2751664971

#4: cdf_cauchy(1, 2, 0.7) = 0.4526068577

#5: VECTOR(pdf_cauchy(1, s, x), s, [0.25, 0.5, 1, 2])

#6: VECTOR(cdf_cauchy(1, s, x), s, [0.25, 0.5, 1, 2])



pdf and cdf for the Cauchy distribution for *t* = 1 and some values for *s*.

Laplace distribution

It is also sometimes called the double exponential distribution, because it can be thought of as two exponential distributions (with an additional location parameter) spliced together back-to-back.

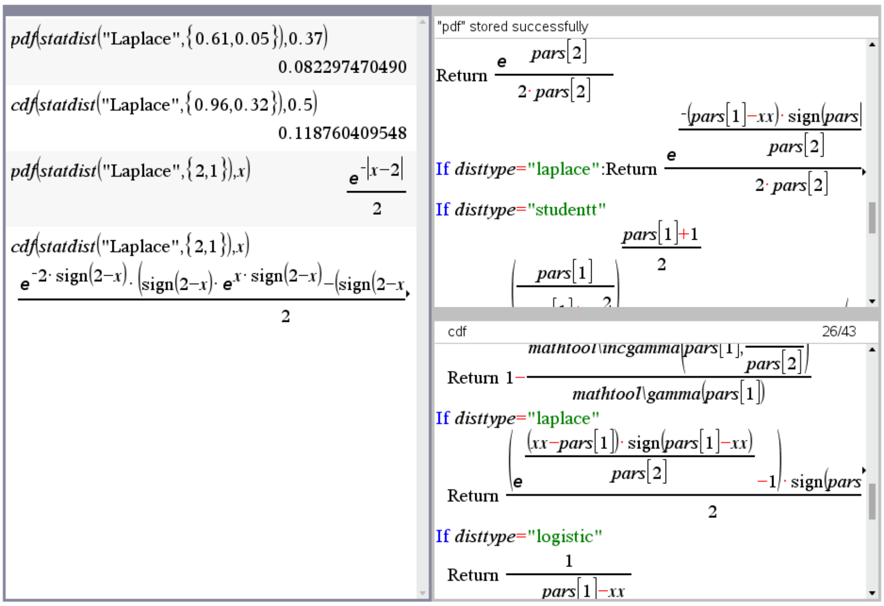
Density function: $f_{\lambda,b}(X)=\frac{1}{2b}\cdot e^{-\frac{|X-\lambda|}{b}}, -\infty < X < \infty,\ b>0,$

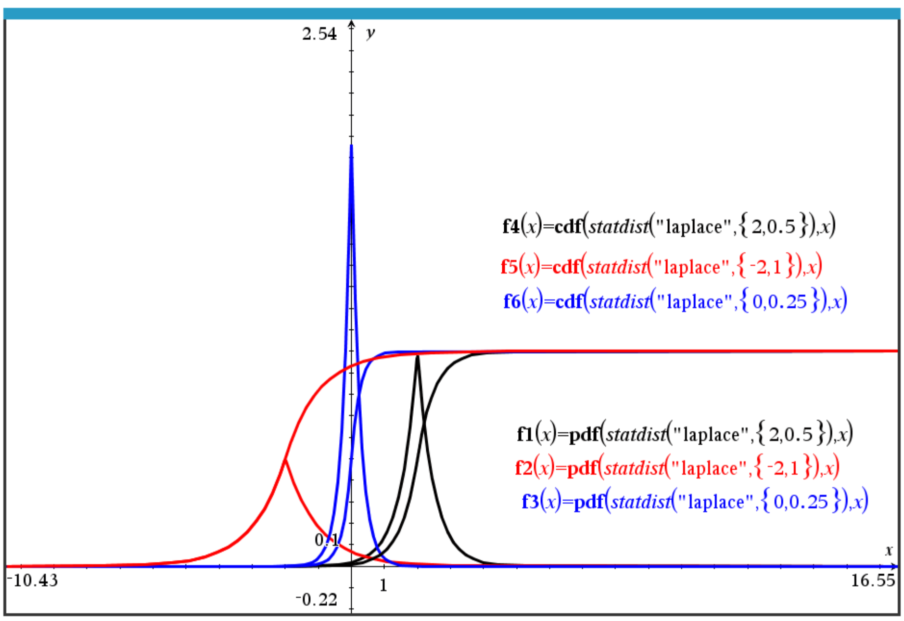
Distribution function: $F_{\lambda,b}(x)=\frac{1}{2}+\frac{1}{2}\cdot \text{sign}(x-\lambda)\cdot\left(1-e^{-\frac{|x-\lambda|}{b}}\right)$

Derive:

$$\text{pdf\_lapl}(\lambda,\ b,\ x) := \frac{1}{2\cdot b}\cdot e^{-|x-\lambda|/b}$$

$$\text{cdf\_lapl}(\lambda,\ b,\ x) := \frac{1}{2}+\frac{1}{2}\cdot \text{SIGN}(x-\lambda)\cdot(1-e^{-|x-\lambda|/b})$$





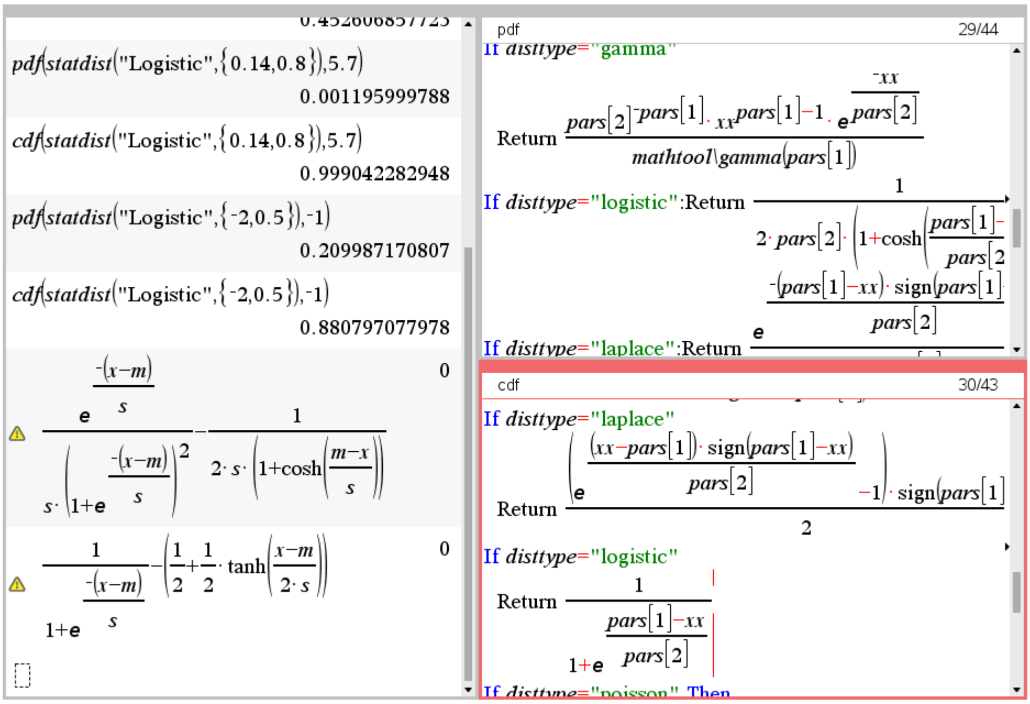
Logistic distribution

The logistic distribution appears in logistic regression and feedforward neural networks. It resembles the normal distribution in shape but has heavier tails (higher kurtosis).
The United States Chess Federation and FIDE have switched its formula for calculating chess ratings from the normal distribution to the logistic distribution; see the article on Elo rating system (itself based on the normal distribution).

Density function: $f_{m,s}(X)=\dfrac{e^{-\frac{X-m}{s}}}{s\left(1+e^{\frac{-(X-m)}{s}}\right)^2}=\dfrac{1}{2s\left(1+\cosh\left(\frac{m-X}{s}\right)\right)},\quad -\infty<X,m<\infty,\ s>0,$

Special case: $f_{0,1}(X)=\dfrac{e^{-X}}{\left(1+e^{-X}\right)^2}$ (= logistic function!)

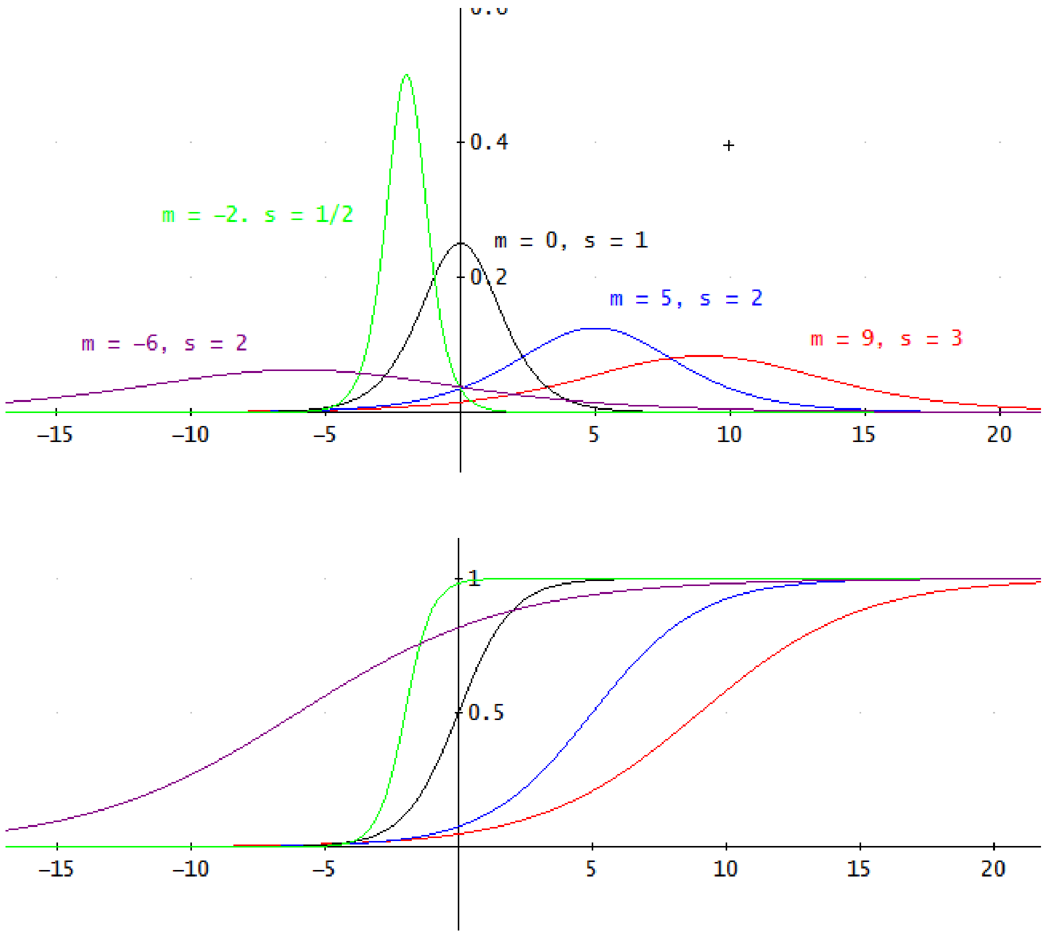
Distribution function: $F_{m,s}(x)=\dfrac{1}{1+e^{-\frac{x-m}{s}}}=\dfrac{1}{2}+\dfrac{1}{2}\tanh\left(\dfrac{x-m}{2s}\right)$



Derive:

$$\text{pdf\_log}(m, s, x) := \frac{e^{-(x-m)/s}}{s\cdot(1+e^{-(x-m)/s})^2}\qquad \text{cdf\_log}(m, s, x) := \frac{1}{1+e^{-(x-m)/s}}$$

$$\text{VECTOR}\left(\text{pdf\_log}(v_1, v_2, x), v, \begin{bmatrix} 9 & 3\\ 5 & 2\\ 0 & 1\\ -2 & \frac{1}{2}\\ -6 & 4\end{bmatrix}\right)\qquad \text{VECTOR}\left(\text{cdf\_log}(v_1, v_2, x), v, \begin{bmatrix} 9 & 3\\ 5 & 2\\ 0 & 1\\ -2 & \frac{1}{2}\\ -6 & 4\end{bmatrix}\right)$$



Same color – same distribution

Poisson distribution

The Poisson distribution expresses the probability of a given number of events occurring in a fixed interval of time or space if these events occur with a known constant mean rate and independently of the time since the last event.

Density function: $f_{\lambda,k}(X=k)=\dfrac{\lambda^k \cdot e^{-\lambda}}{k!}, 0<\lambda<\infty,\ k\in\mathbb{N}_0$

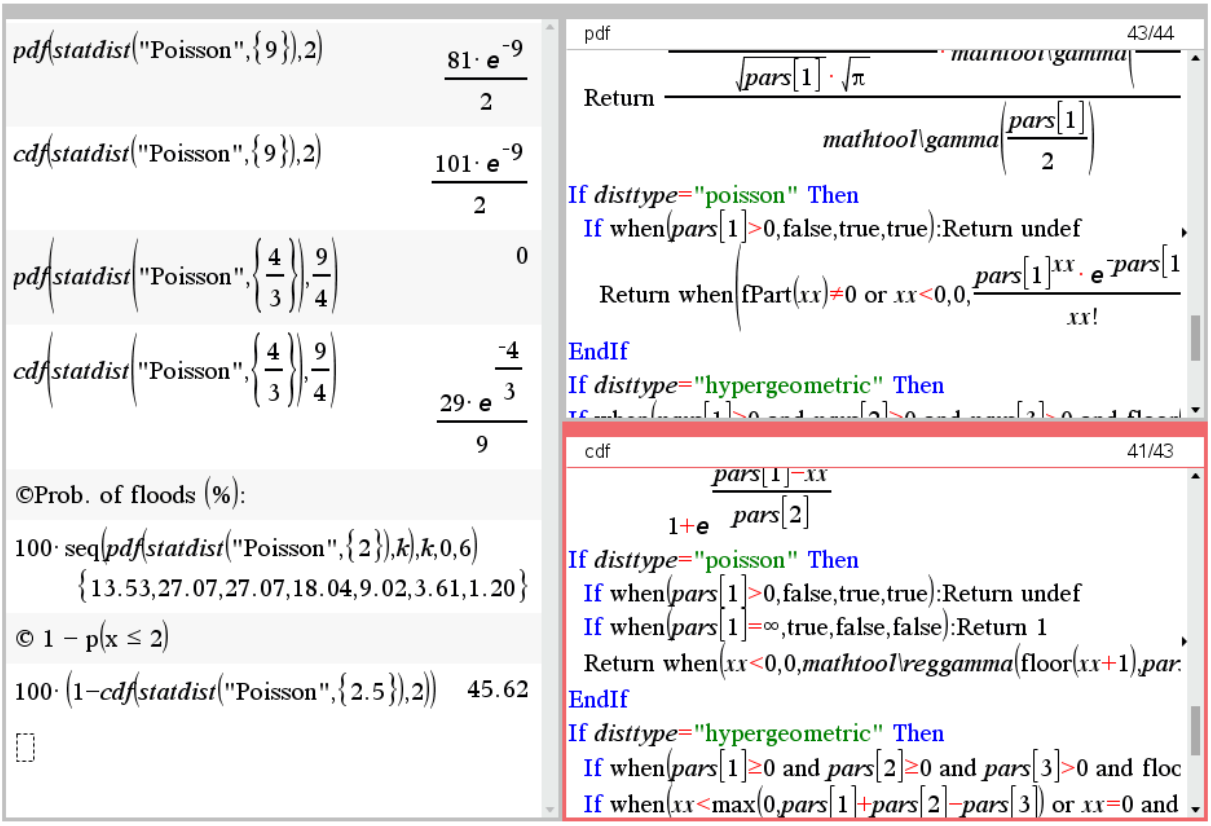
Distribution function: $f_{\lambda,k}(x\le k)=e^{-\lambda}\cdot\sum_{i=0}^{\lfloor k\rfloor}\dfrac{\lambda^i}{i!}$

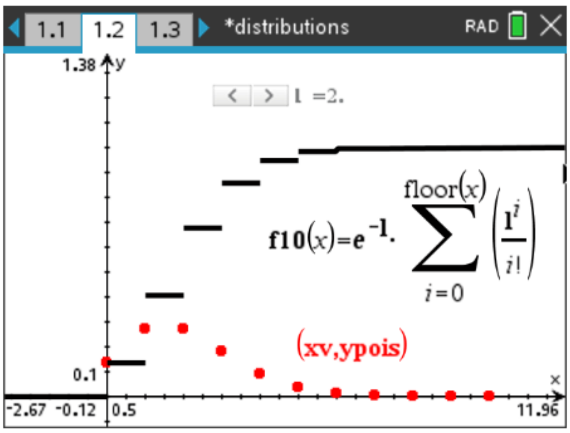
Two examples:

On a particular river, overflow floods occur once every 50 years on average. Calculate the probability of k = 0, 1, 2, 3, 4, 5, or 6 overflow floods in a 100-year interval, assuming the Poisson model is appropriate. Because the average event rate is one overflow flood per 100 years, $\lambda = 2$

Ugarte and colleagues report that the average number of goals in a World Cup soccer match is approximately 2.5 and the Poisson model is appropriate. Because the average event rate is 2.5 goals per match, $\lambda = 2.5$. What is the probability to have at least 3 goals in a WC soccer match?

Answers are given in the TI-Nspire-screen shot below.





This distribution is implemented in *DERIVE*.

A slider can be used to visualize density and distribution function for various values of λ.

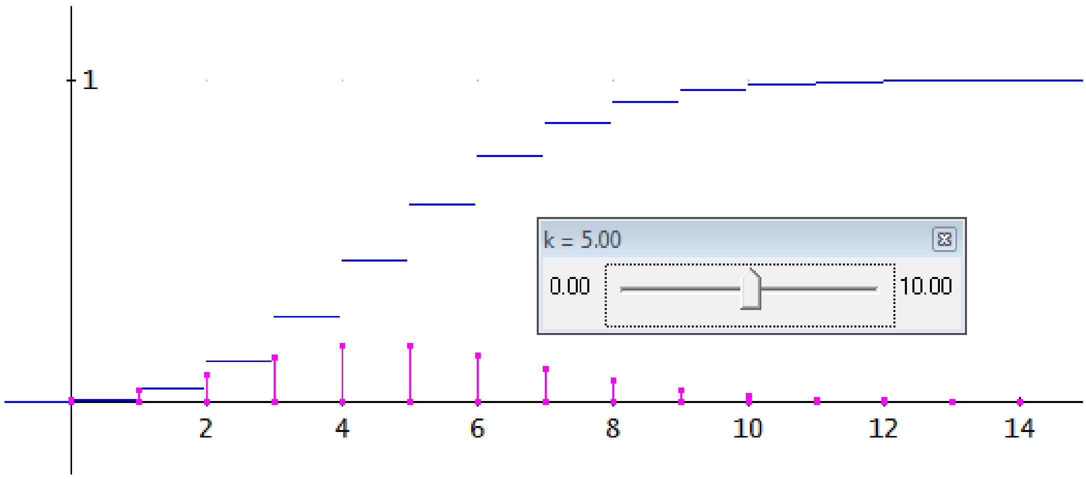
$$\text{POISSON\_DENSITY}(2,\ 9) = \frac{81\cdot e^{-9}}{2}$$

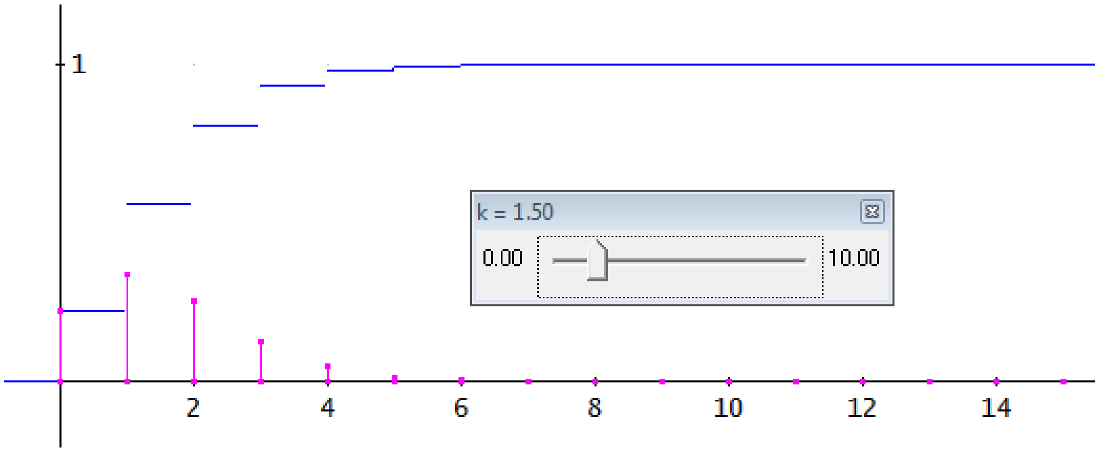
$$\text{POISSON\_DISTRIBUTION}(2,\ 9) = \frac{101\cdot e^{-9}}{2}$$

$$\text{POISSON\_DISTRIBUTION}\left(\frac{9}{4},\ \frac{4}{3}\right) = \frac{29\cdot e^{-4/3}}{9}$$

$$\text{VECTOR}\left(\begin{bmatrix} x & 0 \\ x & \text{POISSON\_DENSITY}(x,\ k) \end{bmatrix},\ x,\ 0,\ 20\right)$$

POISSON_DISTRIBUTION(x, k)





Two *Derive* plots of Poisson pdfs with different parameters.

https://en.wikipedia.org/wiki/Poisson_distribution

https://de.wikipedia.org/wiki/Poisson-Verteilung

https://www.sciencedirect.com/topics/mathematics/poisson-distribution

https://towardsdatascience.com/the-poisson-distribution-and-poisson-process-explained-4e2cb17d459

https://brilliant.org/wiki/poisson-distribution/

You can find many valuable resources in the internet. A good reference for results is WxMaxima which has many of the distributions implemented – and not only the well-known ones.

Very useful are the *MATHEMATICA* websites:

https://www.wolframalpha.com/

https://mathworld.wolfram.com/

Hubert and Wilfried were inspired to work on a problem appearing in the Bolyai Competition from 2021. The German version of this article contains a second problem and can be downloaded (see references). All programs are included in MTH122.zip. Josef

Interplay between Simulation and Theory

Hubert Langlotz & Wilfried Zappe, GER

9. Wir würfeln mit einem nicht gefälschten Würfel (Laplace-Würfel) so lange, bis die Summe *S* der gewürfelten Zahlen 100 übersteigt. Welcher Wert von den gegebenen ist die wahrscheinlichste Summe?

(A) *101* **(B)** *102* **(C)** *103* **(D)** *104* **(E)** *105*

9. We roll with a non-fake dice (Laplace dice) as long, until the sum S of the numbers rolled exceeds 100. Which value of the given ones is the most probable sum?

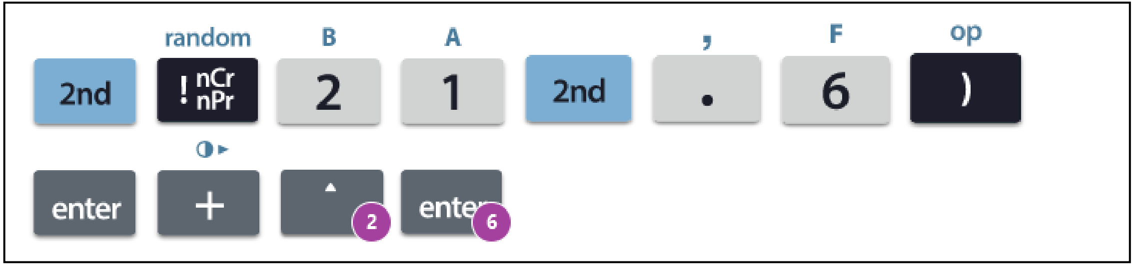
(A) *101* **(B)** *102* **(C)** *103* **(D)** *104* **(E)** *105*

What was exciting about the task was that many students as well as teachers tried to argue with the expected value (3.5) and gave 101, 102 and 105 as the most probable solution. Others argued that all numbers were equally probable, since it is a Laplace cube with equal probabilities of 1-6 and thus also those of 101-106.
So here it was obvious to try simulations. A first approach without programming is, for example, again conceivable with a spreadsheet.

1. Simulations with a scientific pocket calculator random numbers

With randint(1, 6) a "die number" is generated from the interval [1; 6] with the calculator. Then one continues with "the last answer *ans*". Press the "+" key and then copy the randint(1,6) command from line 1 to line 2 (or type it again). Then all what to do is pressing ENTER repeatedly to generate the cumulative sum of the die numbers until the sum exceeds 100 for the first time.

Key sequence:



Possible Tasks:

Write down the intermediate results in the table (partner work):

Seq 1	Seq 2	Seq 3	Seq 4	Seq 5	Seq 6	Seq 7	Seq 8	Seq 9	Seq 10
Note: You should prepare a worksheet with about 35 lines.									

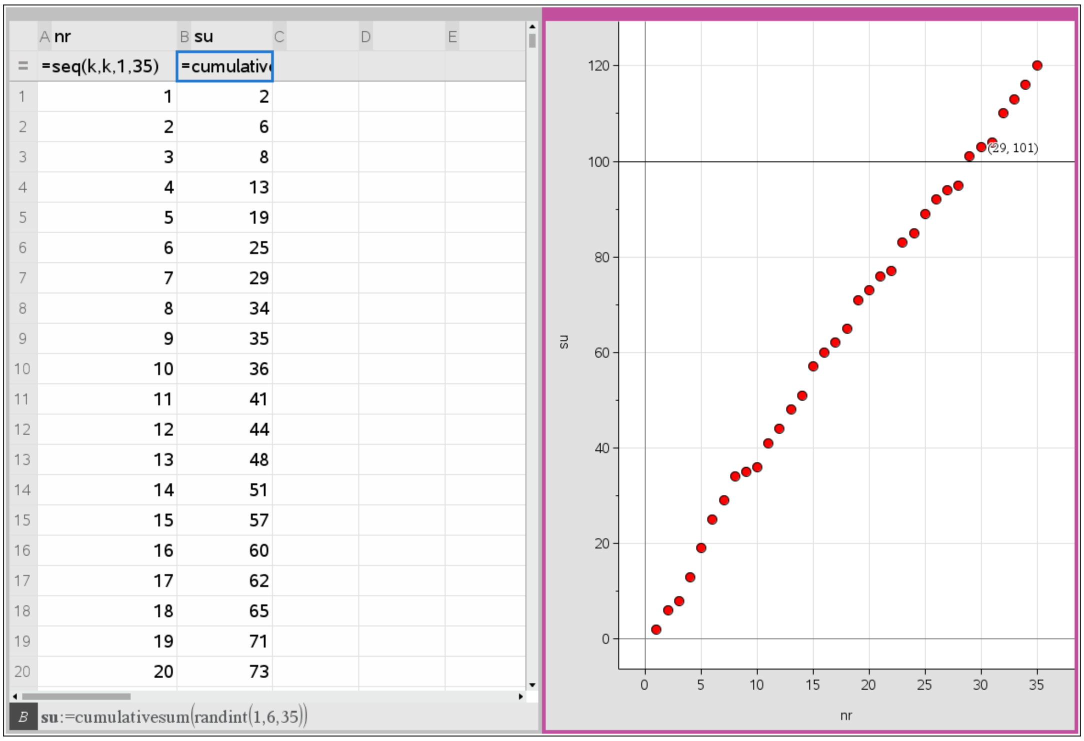
a) In partner work, determine the relative frequencies for the sums 101 to 105.

b) Summarize the results of the class. Do you have any conjecture?

c) Which numbers can be used as penultimate sum before exceeding 100?

d) For example, if the last number before exceeding 100 is 95, what is the probability of reaching 101, 102, 103, 104, 105 as a sum with the next roll?

2. Simulation with the spreadsheet of the CAS calculator.



Possible Tasks:

a) Describe the simulation shown on the screenshots in the context of the issue under investigation.

b) Create a file according to the example on the screenshots to determine the cumulative sums of the cube numbers.

c) Working in pairs, determine which of the sums 101 to 105 is reached first after the subtotal 100 is exceeded. Realize at least 30 repetitions of this simulation with Ctrl+R, note the results and create a frequency distribution.

Sum					
# of throws					

d) Summarize the results of the class.

The simulation methods with a scientific calculator or the method described above with the spreadsheet have the advantage that the process to be observed becomes more "tangible". However, reasonably large test scopes can only be achieved approximately by combining the results of individual tests with individual student work or partner work.

3. Programming with TI-Basic (see also bolyai_e.tns)

Parameter *n* defines the number of simulations. Parameter *zz* can be used to set any limit that must be exceeded.

(In the task considered here *zz* = 100).

List "*numbs*" contains the target numbers, for *zz* = 100 holds:
numbs = {101,102,103,104,105,106}.

Before each of the *n* runs, variables *s* and *j* are set to zero. Variable *s* is increased by one cube number until the limit entered at the start of the program (here *zz* = 100) is exceeded for the first time.

The while-loop terminates when the bound is exceeded. Counter *j* counts the number of steps needed to reach one of the target numbers when the while loop is run once. In the list *counter* is counted, which target number was reached during the run of a while-loop.

```
bolyai_e                                   10/16
Define bolyai_e(n,zz)=
Prgm
Local ev,zz,i,v,nmbs,counter,j,s
ev:=0
numbs:={zz+1,zz+2,zz+3,zz+4,zz+5,zz+6}
counter:={0,0,0,0,0,0}
For i,1,n
  s:=0:j:=0
  While s≤zz
   j:=j+1:v:=randInt(1,6):s:=s+v
  EndWhile
  counter[s-zz]:=counter[s-zz]+1
  ev:=ev+j
EndFor
ev:=ev/n
Disp numbs
Disp counter
Disp 1.·ev
EndPrgm
```

counter[*s-zz*] controls the list element with the number *s-zz*. After *n* while loops, list *counter* indicates how often each target number was reached. By means of the two lists *numbs* and *counter* the output is made possible. They can be used to produce a scatter diagram (see below: it looks the same with TI-BASIC and with TI-Python).

Variable *ev* gives the average number of throws to reach the goal.

```
bolyai_e(1000,100)
    {101,102,103,104,105,106}
    {265,247,208,138,98,44}
    29.326
                                  Done
bolyai_e(10000,100)
    {101,102,103,104,105,106}
    {2861,2385,1870,1426,961,497}
    29.3206
                                  Done
```

The TI-Basic program has the advantage that a sufficient number of simulations can be carried out much faster than with the methods described so far in order to arrive at better justifiable assumptions.

However, the TI-Basic program has the disadvantage that the number of experiments should not be set above n = 10 000.

Already with n = 10 000 the program on the PC needs more than 17 seconds compared to less than one second with the Python program presented below. (*I did it much faster, Josef*)

4. Programming with Python

The big advantage of this small Python program compared to the spreadsheet and the Basic program is that it is possible to run 1 000 000 trials without any problems (duration less than six seconds), a disadvantage could be an error in the algorithm, of course.

Parameter *n* in the function *throws*() defines the number of simulations. Parameter *zz* can be used to set arbitrary bounds that have to be crossed.

(In the actual task *zz* = 100.)

The output is made possible by means of the two lists *numbs* and *counter*.

The while loop terminates when the barrier is exceeded. Variable *ev* presents the average number of throws.

The actual experiment is then performed via the function *experiment*(), in which the function *throws*() defined above is called.

The output is realized via the two print commands in the last for-loop.

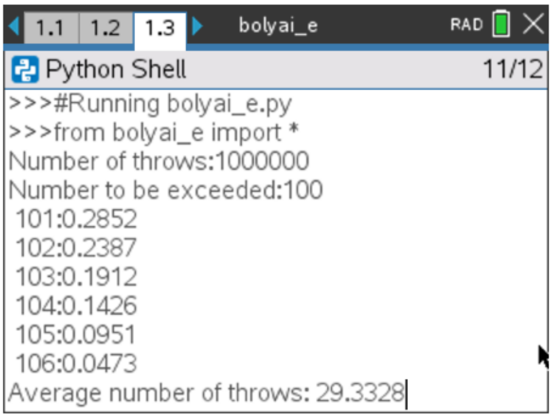
```
bolyai_e.py saved successfully
from random import *
from ti_system import *
def throws(n,zz):
  ev=0
  numbs=[zz+1,zz+2,zz+3,zz+4,zz+5,zz+6]
  counter=[0,0,0,0,0,0]
  for i in range(n):
    s=0
    j=0
    while s<=zz:
      j=j+1
      v=randint(1,6)
      s=s+v
    counter[s-zz-1]+=1
    ev=ev+j
  ev=ev/n
  store_list("numbs",numbs)
  store_list("counter",counter)
  store_value("ev",ev)
```

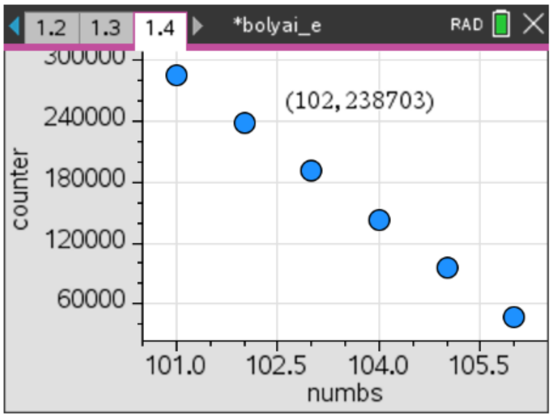
Program continued:

```
def experiment():
    n=int(input("Number of throws:"))
    zz=int(input("Number to be exceeded:"))
    throws(n,zz)
    numbs=recall_list("numbs")
    counter=recall_list("counter")
    ev=Var=recall_value("ev")
    for j in range(6):
        print("%4u:"%numbs[j]+"%1.4f"% float(counter[j]/n))
    print("Average number of throws:",ev)
```

A sample run (Ctrl +R):

```
experiment()
```





Many thanks to Veit Berger for his support producing the Python-program.

Performing 1 million attempts (running time of the program on the PC approx. 6 seconds) the results are as stable as indicated above, i.e. in approx. 28.5% of the attempts 101 is the "final number".

How could this now be justified?

It becomes relatively clear when inspecting the probabilities of getting to these numbers if one has previously reached one of the numbers 95, 96, ...,99, 100.

	101	102	103	104	105	106
95	$\frac{1}{6}$					
96	$\frac{1}{6}$	$\frac{1}{6}$				
97	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$			
98	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$		
99	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$		
100	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$	$\frac{1}{6}$

Six of the 21 cases (about 28.5%) arrive at 101, 5 of the 21 cases (about 23.8%) arrive at 102, and so on. Thus, the Bolyai task seems to be solved.

However, an essential objection is: If we do it this way, then we assume the "equal probability" of reaching the numbers 95, 96, ..., 99, 100.

Can this be justified?

Andreas Eichler and Frank Förster give explanations in their very nice article “Ein Märchenspiel – Stochastische Modellbildung bei einem merkwürdigen Brettspiel“ (see Istronheft Bd. 12, S. 107 ff.) which they justify by simulation and mathematical considerations as well. They show why the "alternating hypothetical" limit for the probability of hitting any number is the inverse of the expected value of the single die roll, namely 2/7 (~ 28.57%). See the values achieved by recursive calculation for the numbers from 1 to 100.

Feld	P(Z = m)	Feld	P(Z = m)	Feld	P(Z = m)	Feld	P(Z = m)	Feld	P(Z = m)
1	0,166667	21	0,285968	41	0,285714	61	0,285714	81	0,285714
2	0,194444	22	0,285944	42	0,285714	62	0,285714	82	0,285714
3	0,226852	23	0,285756	43	0,285715	63	0,285714	83	0,285714
4	0,264660	24	0,285598	44	0,285715	64	0,285714	84	0,285714
5	0,308771	25	0,285600	45	0,285714	65	0,285714	85	0,285714
6	0,360232	26	0,285748	46	0,285714	66	0,285714	86	0,285714
7	0,253604	27	0,285769	47	0,285714	67	0,285714	87	0,285714
8	0,268094	28	0,285736	48	0,285714	68	0,285714	88	0,285714
9	0,280369	29	0,285701	49	0,285714	69	0,285714	89	0,285714
10	0,289288	30	0,285692	50	0,285714	70	0,285714	90	0,285714
11	0,293393	31	0,285707	51	0,285714	71	0,285714	91	0,285714
12	0,290830	32	0,285725	52	0,285714	72	0,285714	92	0,285714
13	0,279263	33	0,285722	53	0,285714	73	0,285714	93	0,285714
14	0,283540	34	0,285714	54	0,285714	74	0,285714	94	0,285714
15	0,286114	35	0,285710	55	0,285714	75	0,285714	95	0,285714
16	0,287071	36	0,285712	56	0,285714	76	0,285714	96	0,285714
17	0,286702	37	0,285715	57	0,285714	77	0,285714	97	0,285714
18	0,285587	38	0,285716	58	0,285714	78	0,285714	98	0,285714
19	0,284713	39	0,285715	59	0,285714	79	0,285714	99	0,285714
20	0,285621	40	0,285714	60	0,285714	80	0,285714	100	0,285714

Tabelle 3: Exakte Werte der rekursiven Berechnung für das leere Spielfeld

112 Andreas Eichler & Frank Förster

Only this justifies the approach of assuming an "equal probability" of reaching the fields 95 - 100.

Note: Use the attached Python program to check this - here e.g. the number 7. It is also possible to print the entire table or parts of it.

```
Python Shell
>>>#Running bolyai_e.py
>>>from bolyai_e import *
Number of throws:1000000
Number to be exceeded:6
  7:253814
```

```
Number of throws: 1000
Lower Bound: 93
Upper Bound: 100
 93 : 0.2660
 94 : 0.2910
 95 : 0.2920
 96 : 0.2710
 97 : 0.3230
 98 : 0.2910
 99 : 0.2710
100 : 0.2800
>>>
```

Solution of the competition organizer:

We denote the sum of the numbers rolled before the last throw with S_1. Because $S_1 \leq 100$ and $s > 100$, $S_1 \leq 95$ must hold. Thus, S_1 can take six different values.

1. $S_1 = 100$: With the same probability s can take 101, 102, …, 106 sein, depending whether the last roll was 1, 2, 3, 4, 5, or 6.
2. $S_1 = 99$: The last roll could be 2, 3, 4, 5 or 6, so values 101, 102, 103, 104, 105 for S are equally probable.
3. $S_1 = 98$: The last roll could be 3, 4, 5 or 6, so S = 101, 102, 103 or 104 are equally probable.

We proceed similarly in the other cases. For $S_1 = 95$ the last roll can only be 6 and the value for S only 101. We can easily see that for S the value 101 is most probable. More precisely: the probability for 101 is with the probability for S1 = 95 greater than the probability for 102 etc. The probabilities decrease in the order 101, 102, 103, 104, 105, 106.

Correct answer: (A)

Comment of the editor: As this is the “D”NL I could not resist adding a *DERIVE*-version of the simulation program. There is one advantage: `dummy:=RANDOM(0)` ensures to get different unpredictable random numbers at each start of the program, no randseed or similar is necessary. Josef

```
bolyai(n, zz, v, ev, i, s, numbs, counter, stats, dummy) :=
  Prog
    dummy := RANDOM(0)
    numbs := [zz + 1, zz + 2, zz + 3, zz + 4, zz + 5, zz + 6]
    counter := [0, 0, 0, 0, 0, 0]
    i := 1
    s := 0
    ev := 0
    Loop
      If i > n exit
      s := 0
      Loop
        If s > zz exit
        ev := ev + 1
        v := RANDOM(6) + 1
        s := s + v
        " DISPLAY([s, v])"
      counter↓(s - zz) := counter↓(s - zz) + 1
      i := i + 1
    RETURN [numbs, counter/n·100, ["", "", "", "", "AVG", APPROX(ev/n)]]
```

bolyai(10000, 100)

101	102	103	104	105	106
27.75	23.71	19.38	14.79	9.76	4.61
				AVG	29.3469

bolyai(100000, 100)

101	102	103	104	105	106
28.672	23.861	18.879	14.241	9.476	4.871
				AVG	29.32695

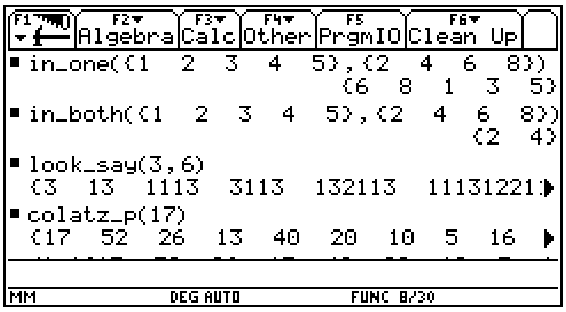
bolyai(10000, 500)

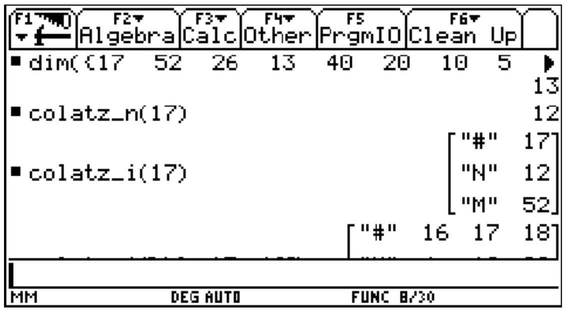
501	502	503	504	505	506
28.76	24.16	18.67	14.2	9.34	4.87
				AVG	143.5003

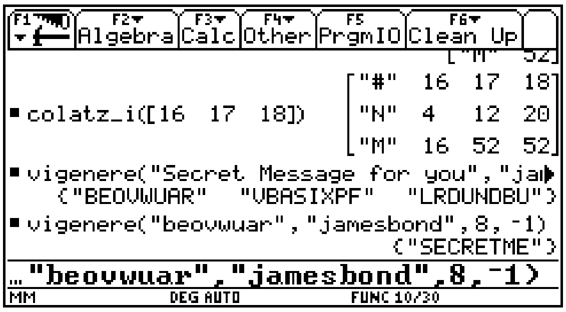
References:

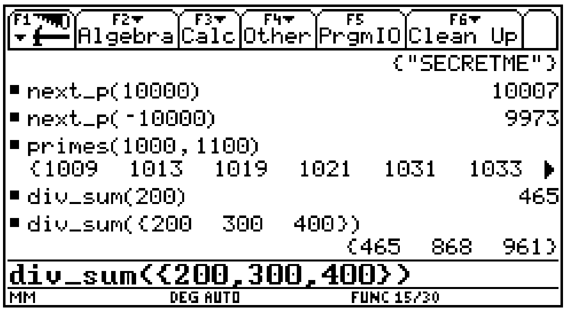
https://ti-unterrichtsmaterialien.net/fileadmin/DE-Materialien/Bolyai_2021.pdf
https://www.bolyaiteam.de/

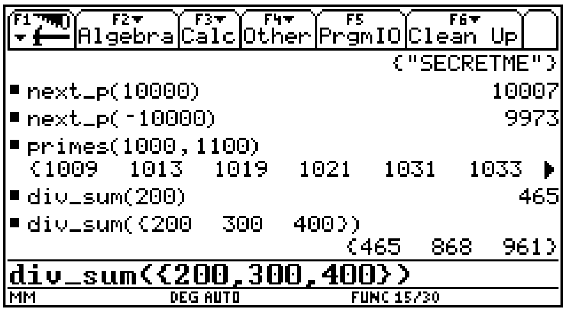
It took me some time to type Matthew's functions based on printed and/or hand written code into my handheld (Voyage 200). Here are some screen shots. Josef

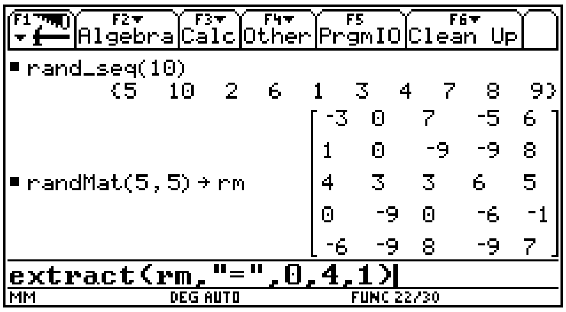


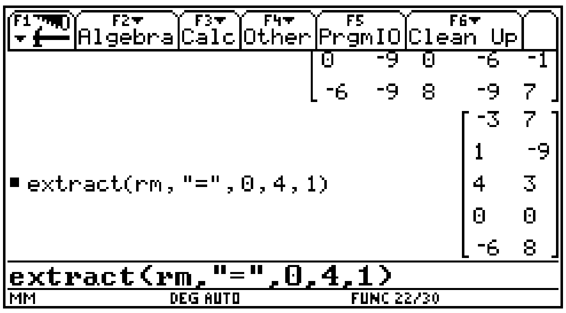


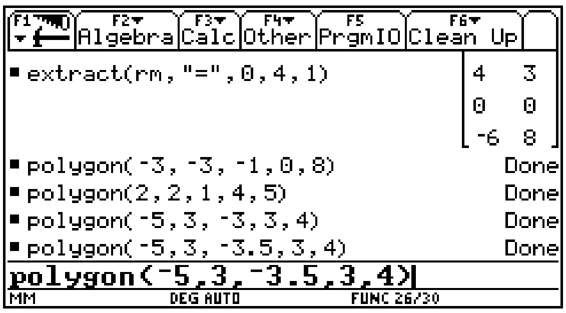


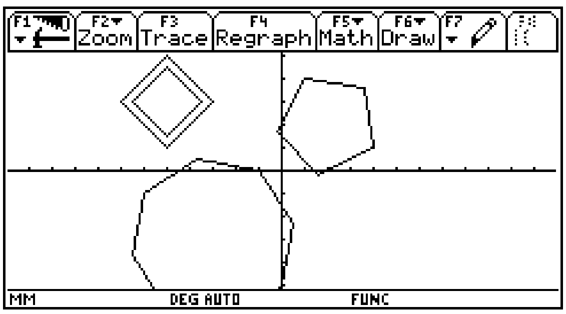








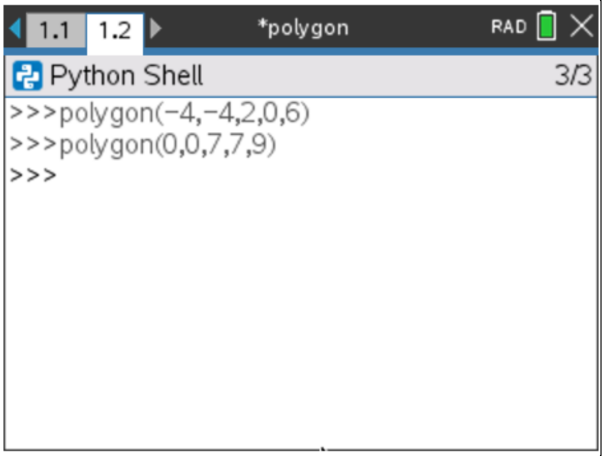


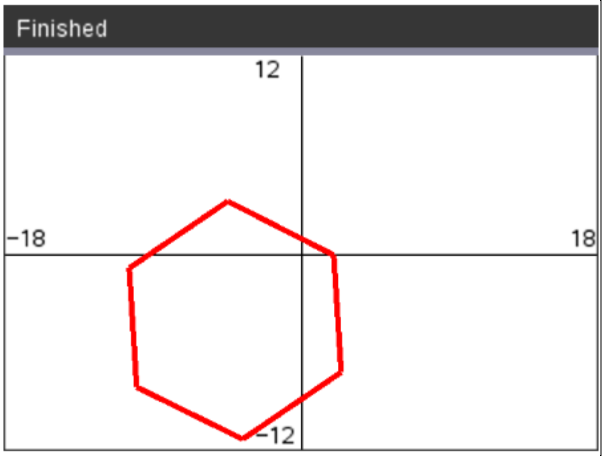


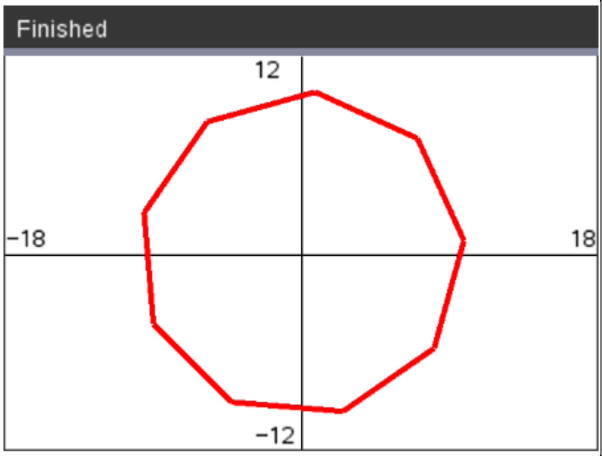
As I mentioned above, we cannot program the Graphs-Application using TI-BASIC on TI-NspireCAS. TI-Python allows to do this. I am not a python expert at all, but I tried to transfer Matthew's TI-89 program to TI-Python. I didn't change the structure of the program and used the same variable names. What I did was skipping the control procedures for entering correct values for the variables.

polygon.py

```
from math import *
sqrt,sin,cos,asin,degrees,radians
import ti_plotlib as plt
def polygon(va,vb,vc,vd,ve):
  vl=sqrt((va-vc)**2+(vb-vd)**2)
  vk=vd-vb
  if vk==0:
    if vc-va<0:
      vm=pi
    else:
      vm=0
  else:
    vm=asin(vk/vl)
    if vc-va<0:
      vm=pi-vm
  vg=2*pi/ve
  vh=vc
  vi=vd
  plt.pen("thin","solid")
  plt.window(-18,18,-12,12)
  plt.color(0,0,0)
  plt.axes("on")
  plt.color(255,0,0)
  plt.pen("medium","solid")
  for vf in range(ve+1):
    vj=va+vl*cos(vm+vg*vf)
    vk=vb+vl*sin(vm+vg*vf)
    plt.line(vh,vi,vj,vk,"default")
    vh=vj
    vi=vk
plt.show_plot()
```







But there is one restriction, though. Plotting works only in handheld mode when working on the PC.