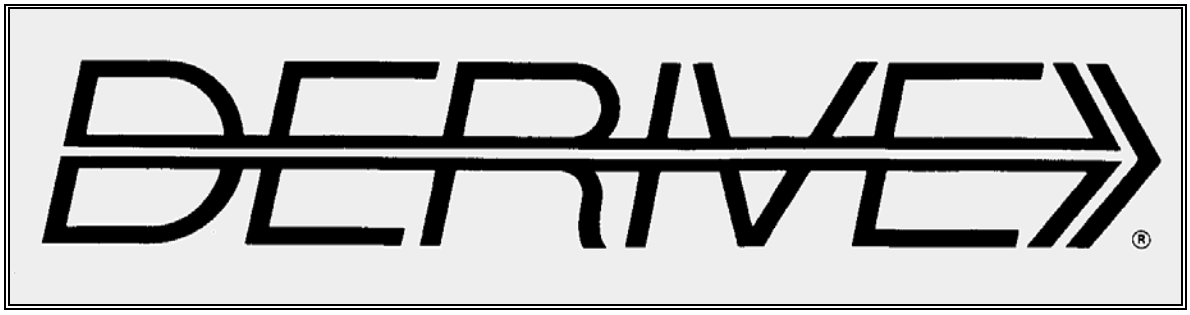
THE DERIVE - NEWSLETTER #111

ISSN 1990-7079

THE BULLETIN OF THE



USER GROUP

+ CAS-TI

Contents:

ÉTS will host ACA 2019

Located in the heart of downtown Montreal and in full growth, l'École de technologie supérieure (**ÉTS**) is pleased to host the 25th Conference on *Applications of Computer Algebra* (**ACA**) to be held from **July 16 to July 20, 2019**.

The **ACA** conference is devoted to promoting computer algebra applications as well as foster interaction and exchange among developers of computer algebra systems with researchers and users. The latter are mostly scientists, engineers, educators and mathematicians.

Call for Sessions

The **ACA** conference sessions are usually held simultaneously and are independent of any specific product or software. In each session, presenters hold a 30 minute or a one hour talk with an objective to present research problem-solving using symbolic computation as well as applications of computer algebra in research. Subjects vary and include the fields of engineering, sciences, health and medicine, applied and pure mathematics, business and finance as well as data processing.

Mathematics, science and business teachers using symbolic computation in their teaching may be interested by the **Computer Algebra in Education** session.

Important dates

Deadline for session proposals : **February 8, 2019**

Deadline for submission of abstracts : **April 19, 2019**

Registration deadline *(early bird)* : **May 31, 2019**

Committees

Organizing Committee

Louis-Xavier Proulx (chair), **Michel Beaudin**, **Anouk Bergeron-Brlek**, **Richard Rioux**, École de technologie supérieure

Program Chairs

Michel Beaudin, École de technologie supérieure
Michael Wester, University of New Mexico

aca2019.etsmtl.ca

It would be great to meet many members of the DERIVE- and TI-Community in Montréal next summer. I am sure that Michel and his team will organize an excellent event. I can recommend participating at an ACA-conference because it offers one of the rare opportunities for us math-teachers to get rich information and insight about other CAS-applications besides "only" math teaching.

Dear DUG Members,

Welcome to DNL#111. It is now mid-October – a little bit late for the September issue. I waited for some information for ACA 2019.

Now you can find the Announcement of ACA 2019 together with a Call for Sessions on opposite page.

I also wanted to wait for answers on my Nspire-requests (see the User Forum) from the TIers. But it seems to need a lot of time to send answers on users' requests. I hope to receive the respective explanations for the next issue.

Please notice the links given on page 42. There are some useful textbooks free for download as pdf-files. Follow the link to Miroslav's Puzzle Page.

I am very happy for Michel's permission to publish his lecture given at ACA 2018 in La Compostela (Spain) this summer. I couldn't resist to add some "DERIVE-comments".

Wolfgang Alvermann sent an article connected with ship-building. He was teacher at a Technical College in North-Germany. I wanted to leave his German version and did not translate it paragraph for paragraph. I tried another way and translated page for page. Hope you like this.

Finally, you can find the first one of Don Phillips' TI-Nspire Goodies and the first part of Don Speray's series on the sum of Powers, enjoy it.

Best regards and wishes

Josef

We used the wonderful weather to undertake a hiking week in the Dolomites in South Tyrol (Italy). You see parts of the famous "Rosengarten".

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: December2018

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
Tools for 3D-Problems, P. Lüke-Rosendahl, GER
Simulating a Graphing Calculator in *DERIVE*, J. Böhm, AUT
Graphics World, Currency Change, P. Charland, CAN
Cubics, Quartics – Interesting features, T. Koller & J. Böhm, AUT
Logos of Companies as an Inspiration for Math Teaching
Exciting Surfaces in the FAZ / Pierre Charland´s Graphics Gallery
BooleanPlots.mth, P. Schofield, UK
Old traditional examples for a CAS – What´s new? J. Böhm, AUT
Mandelbrot and Newton with *DERIVE*, Roman Hašek, CZK
Tutorials for the NSpireCAS, G. Herweyers, BEL
Some Projects with Students, R. Schröder, GER
Dirac Algebra, Clifford Algebra, D. R. Lunsford, USA
A New Approach to Taylor Series, D. Oertel, GER
Rational Hooks, J. Lechner, AUT
Statistics of Shuffling Cards, Charge in a Magnetic Field, H. Ludwig, GER
Factoring Trinomials, D. McDougall, CAN
Selected Lectures from TIME 2016
More Applications of TI-Innovator™ Hub and TI-Innovator™ Rover
Surfaces and their Duals, Cayley Symmetroid
Affine Mappings –Treated Systematically, H. Nieder, GER
Three planes in space – how they can intersect
Goodies for TI-NspireCAS, Don Phillips, USA
Deriving the formula for the Sum of Power, Don Speray, USA
and others

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

ACA 2018
Applications of Computer Algebra
Session: Computer Algebra in Education
Faculty of Mathematics of the University of Santiago de Compostela
SANTIAGO DE COMPOSTELA, JUNE 18-22, 2018

Michel Beaudin,
ÉTS, Montréal, Canada

Periodic and Nontrivial Periodic Input in Linear ODEs

Overview

1. Preliminary results and cos(ωt) input

Consider the linear second order differential equation

$$m\,y''(t) + b\,y'(t) + k\,y(t) = f(t)$$

where m, b, and k are constants, $m \neq 0$. Here f is a piecewise continuous function.

When m, b and k are *positive*, this ODE represents the motion for the damped spring-mass system with external force $f(t)$. The constants m, b, k are the mass, damping coefficient and spring constant respectively:

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

The following results can be found in a differential equations textbook.

The general solution of

$$m\,y''(t) + b\,y'(t) + k\,y(t) = f(t) \quad (m \neq 0)$$

is $y(t) = y_h(t) + y_p(t)$.

$y_h(t)$ is the *homogeneous solution*, a solution to

$$m\,y''(t) + b\,y'(t) + k\,y(t) = 0.$$

$y_p(t)$ is a *particular solution*, a solution to

$$m\,y''(t) + b\,y'(t) + k\,y(t) = f(t).$$

Set $r_{1,2} = \dfrac{-b \pm \sqrt{b^2 - 4mk}}{2m}$ and $\delta = \dfrac{\sqrt{4mk - b^2}}{2m}$

The solution of the homogeneous equation is:

$$\begin{cases} c_1 e^{r_1 t} + c_2 e^{r_2 t}, & b^2 - 4mk > 0 \\ e^{-\frac{bt}{2m}}\left(c_1 + c_2 t\right), & b^2 - 4mk = 0 \\ e^{-\frac{bt}{2m}}\left(c_1 \cos(\delta t) + c_2 \sin(\delta t)\right), & b^2 - 4mk < 0 \end{cases}$$

where c_1, c_2 are arbitrary constants.

We have chosen m, b and k positive, so the homogeneous solution $y_h(t) \to 0$ when $t \to \infty$ and is often called *transient solution*.

If we add initial conditions to the differential equation, it does not affect the particular solution, only the homogeneous solution. So both the initial position $y(0)$ and the initial velocity $y'(0)$ can be assumed to be 0.

The "deSolve" command of a CAS confirms this: here are some examples:

The following function **output** is using the "deSolve" command of Nspire CX CAS for solving the problem

$$m \cdot \frac{d^2}{dt^2}(y) + b \cdot \frac{d}{dt}(y) + k \cdot y = input,\ y(0) = yo,\ y'(0) = vo$$

output$(m,b,k,input)$:=deSolve$(m \cdot y'' + b \cdot y' + k \cdot y = input$ and $y(0) = yo$ and $y'(0) = vo, t, y)$ ▸ *Done*

Here are some examples illustrating the homogeneous solution converges to 0 when m, b and k are positive and the fact the initial conditions only affect the homogeneous solution (in red), not the particular solution (in green).

output$(1,5,6,t^2)$ ▸ $y = \dfrac{(4 \cdot vo + 12 \cdot yo - 1) \cdot e^{-2 \cdot t}}{4} + \left(-vo - 2 \cdot yo + \dfrac{2}{27}\right) \cdot e^{-3 \cdot t} + \dfrac{t^2}{6} - \dfrac{5 \cdot t}{18} + \dfrac{19}{108}$

output$(1,5,6,e^{-5 \cdot t})$ ▸ $y = \dfrac{(3 \cdot vo + 9 \cdot yo + 1) \cdot e^{-2 \cdot t}}{3} + \left(-vo - 2 \cdot yo - \dfrac{1}{2}\right) \cdot e^{-3 \cdot t} + \dfrac{e^{-5 \cdot t}}{6}$

output$(1,5,6,\cos(7 \cdot t))$ ▸ $y = \dfrac{-43 \cdot \cos(7 \cdot t)}{3074} + \dfrac{35 \cdot \sin(7 \cdot t)}{3074} + \dfrac{(53 \cdot vo + 159 \cdot yo - 2) \cdot e^{-2 \cdot t}}{53} + \left(-vo - 2 \cdot yo + \dfrac{3}{58}\right) \cdot e^{-3 \cdot t}$

output$(1,6,13,\cos(t))$ ▸ $y = \left(yo - \dfrac{1}{15}\right) \cdot e^{-3 \cdot t} \cdot \cos(2 \cdot t) + \dfrac{(30 \cdot vo + 90 \cdot yo - 7) \cdot e^{-3 \cdot t} \cdot \sin(2 \cdot t)}{60} + \dfrac{\cos(t)}{15} + \dfrac{\sin(t)}{30}$

output$(1,8,16,\sin(3 \cdot t))$ ▸ $y = \dfrac{-24 \cdot \cos(3 \cdot t)}{625} + \dfrac{7 \cdot \sin(3 \cdot t)}{625} + \left(\left(vo + 4 \cdot yo + \dfrac{3}{25}\right) \cdot t + yo + \dfrac{24}{625}\right) \cdot e^{-4 \cdot t}$

(You need to have in MyLib the 3 library files files kit_ets_mb.tns, kit_ets_fh.tns and ets_specfunc.tna).

I'd like to reproduce the calculation using DERIVE – no problems can be faced! (Josef). (Beaudin_1.dfw)

$$\#1:\quad output(m,\ b,\ k,\ input) := DSOLVE2\left(\frac{b}{m},\ \frac{k}{m},\ \frac{input}{m},\ t\right)$$

$$\#2:\quad y1(t) := output(1,\ 5,\ 6,\ t^2)$$

$$\#3:\quad y1(t) := c1\cdot e^{-2\cdot t} + c2\cdot e^{-3\cdot t} + \frac{18\cdot t^2 - 30\cdot t + 19}{108}$$

$$\#4:\quad (SOLUTIONS([y1(0) = y\_,\ y1'(0) = v\_],\ [c1,\ c2]))_1$$

$$\#5:\quad \left[\frac{4\cdot v\_ + 12\cdot y\_ - 1}{4},\ -\frac{27\cdot v\_ + 2\cdot(27\cdot y\_ - 1)}{27}\right]$$

$$\#6:\quad y1(t) := \frac{e^{-2\cdot t}\cdot(4\cdot v\_ + 12\cdot y\_ - 1)}{4} - \frac{e^{-3\cdot t}\cdot(27\cdot v\_ + 2\cdot(27\cdot y\_ - 1))}{27} + \frac{18\cdot t^2 - 30\cdot t + 19}{108}$$

$$\#7:\quad y2(t) := output(1,\ 5,\ 6,\ COS(7\cdot t))$$

$$\#8:\quad y2(t) := c1\cdot e^{-2\cdot t} + c2\cdot e^{-3\cdot t} - \frac{43\cdot COS(7\cdot t)}{3074} + \frac{35\cdot SIN(7\cdot t)}{3074}$$

$$\#9:\quad (SOLUTIONS([y2(0) = y\_,\ y2'(0) = v\_],\ [c1,\ c2]))_1$$

$$\#10:\quad \left[\frac{53\cdot v\_ + 159\cdot y\_ - 2}{53},\ -\frac{58\cdot v\_ + 116\cdot y\_ - 3}{58}\right]$$

$$\#11:\quad y2(t) := \frac{e^{-2\cdot t}\cdot(53\cdot v\_ + 159\cdot y\_ - 2)}{53} - \frac{e^{-3\cdot t}\cdot(58\cdot v\_ + 116\cdot y\_ - 3)}{58} - \frac{43\cdot COS(7\cdot t)}{3074} + \frac{35\cdot SIN(7\cdot t)}{3074}$$

1. Preliminary results and cos(*ωt*) input

Now we choose $f(t) = \cos(\omega t)$ with $\omega > 0$. There is no loss of generality to take 1 as the leading coefficient of $\cos(\omega t)$ because the ODE is *linear*.

So for the rest of this part I, we consider the differential equation

$$m\,y''(t) + b\,y'(t) + k\,y(t) = \cos(\omega t).$$

And we recall that m, b and k are positive constants.

Every solution of

$$m\,y''(t) + b\,y'(t) + k\,y(t) = \cos(\omega t)$$

has the form

$$y(t) = y_h(t) + ss(t)$$

We already know the *transient solution* $y_h(t)$ converges to 0 as t goes to infinity. $ss(t)$ is the particular solution

$$a\cos(\omega t) + b\sin(\omega t)$$

called the *steady-state solution*.

The constants α and β depend on m, b, k and ω.

1.1 Finding the steady-state solution

We know the nature of the transient solution $y_h(t)$ depends on the sign of $b^2 - 4mk$. But because m, b, and k are positive, we have, as stated before, $\lim_{t\to\infty} y_h(t) = 0$.

Associated vocabulary: the value

$$2\sqrt{k\,m}$$

is known as the *critical damping* .

We have *overdamping* if $b > 2\sqrt{k\,m}$ and *underdamping* if $b < 2\sqrt{k\,m}$.

Fact: for the expression

$$ss(t) = a\cos(\omega t) + b\sin(\omega t)$$

the amplitude is

$$\sqrt{\alpha^2 + \beta^2}.$$

This comes from the identity ("*trigcollect*")

$$\alpha\cos(\omega t) + \beta\sin(\omega t) = \sqrt{\alpha^2 + \beta^2}\,\operatorname{sign}(\alpha)\cos\left(\arctan\left(\frac{\beta}{\alpha}\right) - \varpi t\right)$$

Note (Josef): TI-Nspire's tcollect does not show any reaction. So, we borrow the TrigCollect Mode of *DERIVE*:

$$\alpha\cdot\mathrm{COS}(\omega\cdot t) + \beta\cdot\mathrm{SIN}(\omega\cdot t)$$

$$\mathrm{Trigonometry} := \mathrm{Collect}$$

$$\sqrt{(\alpha^2 + \beta^2)}\cdot\mathrm{SIGN}(\alpha)\cdot\mathrm{COS}\left(\mathrm{ATAN}\left(\frac{\beta}{\alpha}\right) - t\cdot\omega\right)$$

So, an interesting question is:

Which value of the frequency ω of the external force $\cos(\omega t)$ will produce a maximum amplitude for $ss(t)$?

Because we will be only interested in the steady-state solution of the ODE

$$m\,y''(t) + b\,y'(t) + k\,y(t) = \cos(\omega t)$$

we will use the so-called *method of undetermined coefficients* in order to find it.

Another well-known method (yielding the whole solution) is the *Laplace transform method*. Part II will make use of it.

Finally, the "*deSolve*" command of a CAS can also be used to find the whole solution (the *method of variation of parameters* is used).

The results we will find are summarized in the next slides.

We will show how easy it is to confirm these using Nspire CX CAS.

Moreover, the animation facilities (slider bars) of a CAS can help students for understanding what is going on.

1.2 Frequency repsonse

Let m, b, k and $\omega > 0$. The amplitude of the steady-state solution of the ODE

$$m\,y''(t) + b\,y'(t) + k\,y(t) = \cos(\omega t)$$

is

$$\frac{1}{\sqrt{b^2\varpi^2 + (k - m\omega^2)^2}}.$$

This last expression, as a function of the *frequency* ω, is the amplitude of the steady-state solution which is the *response* to the input $\cos(\omega t)$ as the variable t is increasing.

When $b^2 < 2mk$, the amplitude function

$$\frac{1}{\sqrt{b^2\varpi^2 + (k - m\omega^2)^2}}.$$

reaches a maximum value of $\dfrac{2m}{b\sqrt{4km - b^2}}$ at the point $\omega = \sqrt{\dfrac{k}{m} - \dfrac{b^2}{2m^2}}$.

Note: no interest for $b^2 \geq 2mk$ because the amplitude function is decreasing to 0 from the beginning.

In the classroom, I ask students to take numerical values for m, b and k and confirm the results using the portable CAS calculator. They will need to apply the method of undetermined coefficients, to solve a linear system of equations, to find the maximum value of a function (using both calculus and a graph). Then the general case follows.

Here is an example :

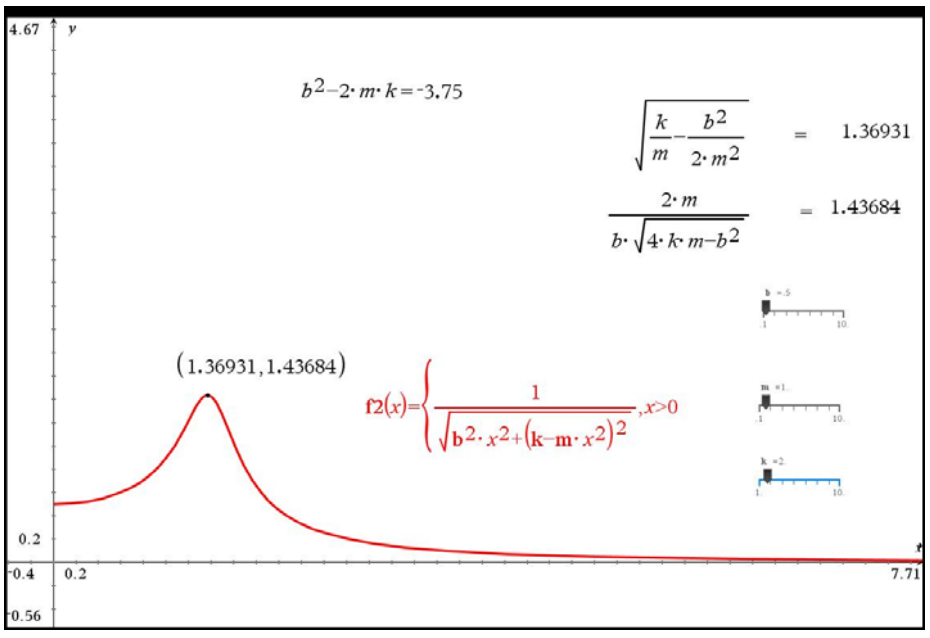
We will take $m = 1$, $b = ½$ and $k = 2$. So $b^2 < 2\ k$.

We will show the frequency

$$\varpi = \sqrt{\frac{1}{2} - \frac{(1/2)^2}{2 \cdot 1^2}} = \frac{\sqrt{30}}{4}$$

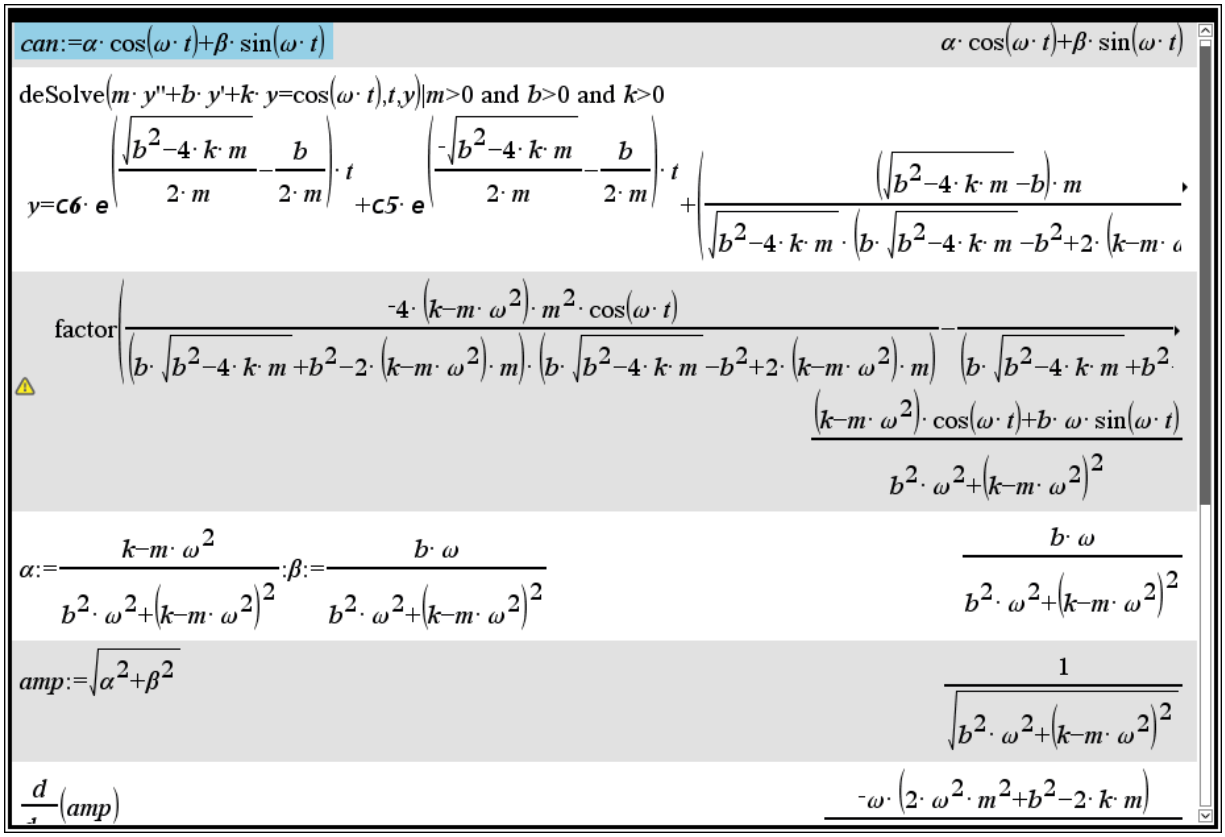
produces a maximum amplitude for the steady-state solution whose value is

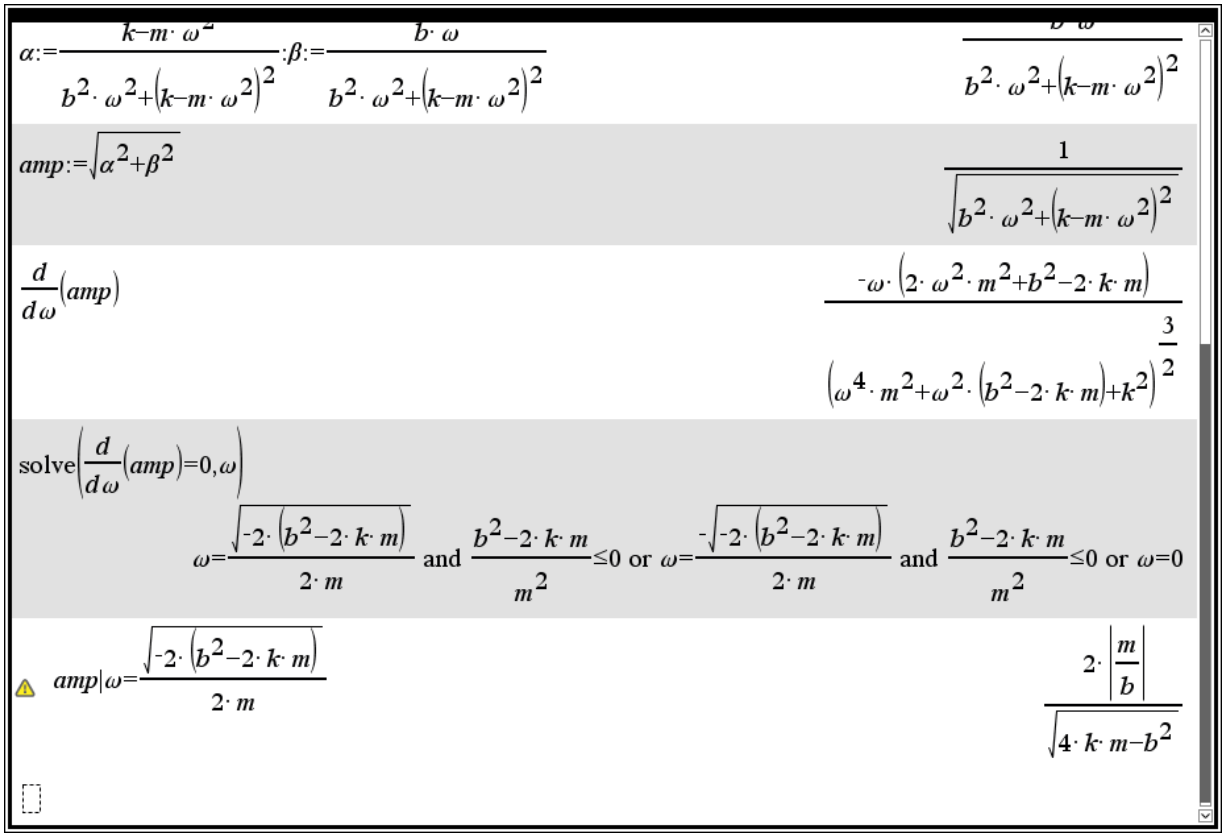
$$\frac{2m}{b\sqrt{4km - b^2}} = \frac{2 \cdot 1}{(1/2)\sqrt{4 \cdot 2 \cdot 1 - (1/2)^2}} = \frac{8\sqrt{31}}{31} \approx 1.44$$



1.3 Demo with Nspire CX CAS

We now move to Nspire CX CAS software in order to perform the example.
Then the software will also be used to prove the general case.





Fortunately I get the same result supported by *DERIVE* (Josef): (Beaudin_2.dfw)

#1: $\text{SOLUTIONS}(m\cdot\lambda^2 + b\cdot\lambda + k = 0,\ \lambda)$

#2: $\left[-\frac{\sqrt{(b^2 - 4\cdot k\cdot m)} + b}{2\cdot m},\ \frac{\sqrt{(b^2 - 4\cdot k\cdot m)} - b}{2\cdot m}\right]$

#3: $\left[r1 := -\frac{\sqrt{(b^2 - 4\cdot k\cdot m)} + b}{2\cdot m},\ r2 := \frac{\sqrt{(b^2 - 4\cdot k\cdot m)} - b}{2\cdot m}\right]$

#4: $yh := \alpha\cdot\text{COS}(\omega\cdot t) + \beta\cdot\text{SIN}(\omega\cdot t)$

#5: $m\cdot\left(\frac{d}{dt}\right)^2 yh + b\cdot\frac{d}{dt} yh + c\cdot yh = \text{COS}(\omega\cdot t)$

#6: $(b\cdot\beta\cdot\omega + c\cdot\alpha - m\cdot\alpha\cdot\omega^2)\cdot\text{COS}(t\cdot\omega) - (b\cdot\alpha\cdot\omega - c\cdot\beta + m\cdot\beta\cdot\omega^2)\cdot\text{SIN}(t\cdot\omega) = \text{COS}(t\cdot\omega)$

#7: $(\text{SOLUTIONS}(\left[b\cdot\beta\cdot\omega + c\cdot\alpha - m\cdot\alpha\cdot\omega^2 = 1,\ b\cdot\alpha\cdot\omega - c\cdot\beta + m\cdot\beta\cdot\omega^2 = 0\right],\ [\alpha,\ \beta]))_1$

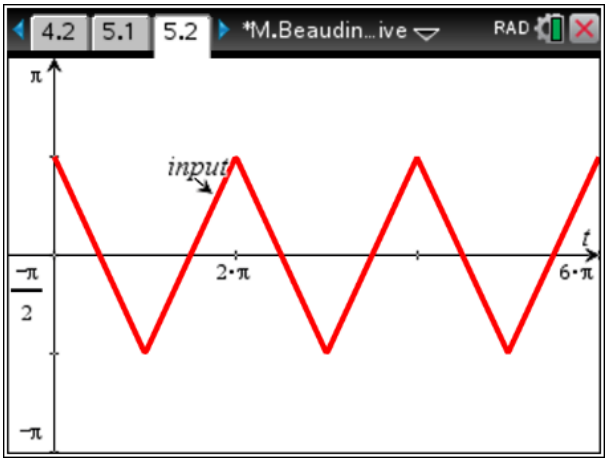
#8: $\left[\frac{c - m\cdot\omega^2}{b^2\cdot\omega^2 + (c - m\cdot\omega^2)^2},\ \frac{b\cdot\omega}{b^2\cdot\omega^2 + (c - m\cdot\omega^2)^2}\right]$

#9: $amp := \left|\left[\frac{c - m\cdot\omega^2}{b^2\cdot\omega^2 + (c - m\cdot\omega^2)^2},\ \frac{b\cdot\omega}{b^2\cdot\omega^2 + (c - m\cdot\omega^2)^2}\right]\right|$

#10: $amp := \frac{1}{\sqrt{(b^2\cdot\omega^2 + (c - m\cdot\omega^2)^2)}}$

2. Nontrivial periodic input

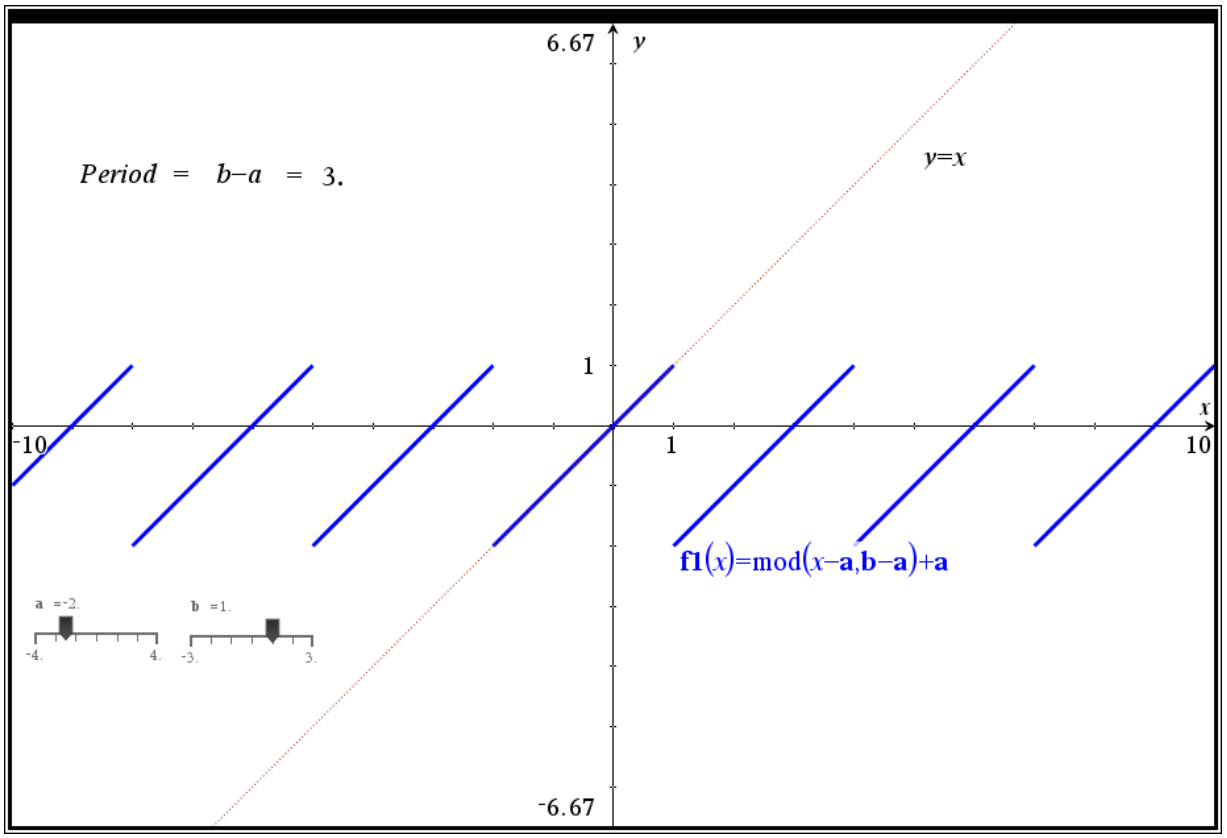
Instead of having a pure cosine as external force, suppose we have an external 2π-periodic input as this one:



At the last ACA in Jerusalem, we used the modulo function of Nspire CAS in order to define a periodic function (the same can also be done with *Derive*).

Let's recall how to do it: if g is a function defined for $a < t < b$, then $g(\text{mod}(t\text{-}a,\ b\text{-}a)+a)$ is the periodic extension of g over the real line of period $b - a$ (see a demonstration below).

So, you only need to define your function over an interval of periodicity and "mod" does the job!



Note: in engineering applications, the nontrivial periodic inputs are usually made of piecewise continuous functions whose right-hand and left-hand derivatives exist at each point of the interval. The former function is such an example:

We can represent this function by its Fourier series: we replace it by the trigonometric series

$$\frac{a_0}{2}+\sum_{n=1}^{\infty}\left(a_n\cos(nt)+b_n\sin(nt)\right)$$

where the coefficients a_0, a_n and b_n are given by

$$a_0=\frac{1}{\pi}\int_0^{2\pi} f(t)\,dt;\ a_n=\frac{1}{\pi}\int_0^{2\pi} f(t)\cos(nt)\,dt;\ b_n=\frac{1}{\pi}\int_0^{2\pi} f(t)\sin(nt)\,dt$$

We showed at ACA 2013 how to program in Nspire CX CAS a « fourier » function yielding any partial sum of a piecewise continuous function. We will make use of it later.

Fact: the Fourier coefficients converge to 0 as n goes to infinity (using integration by parts).

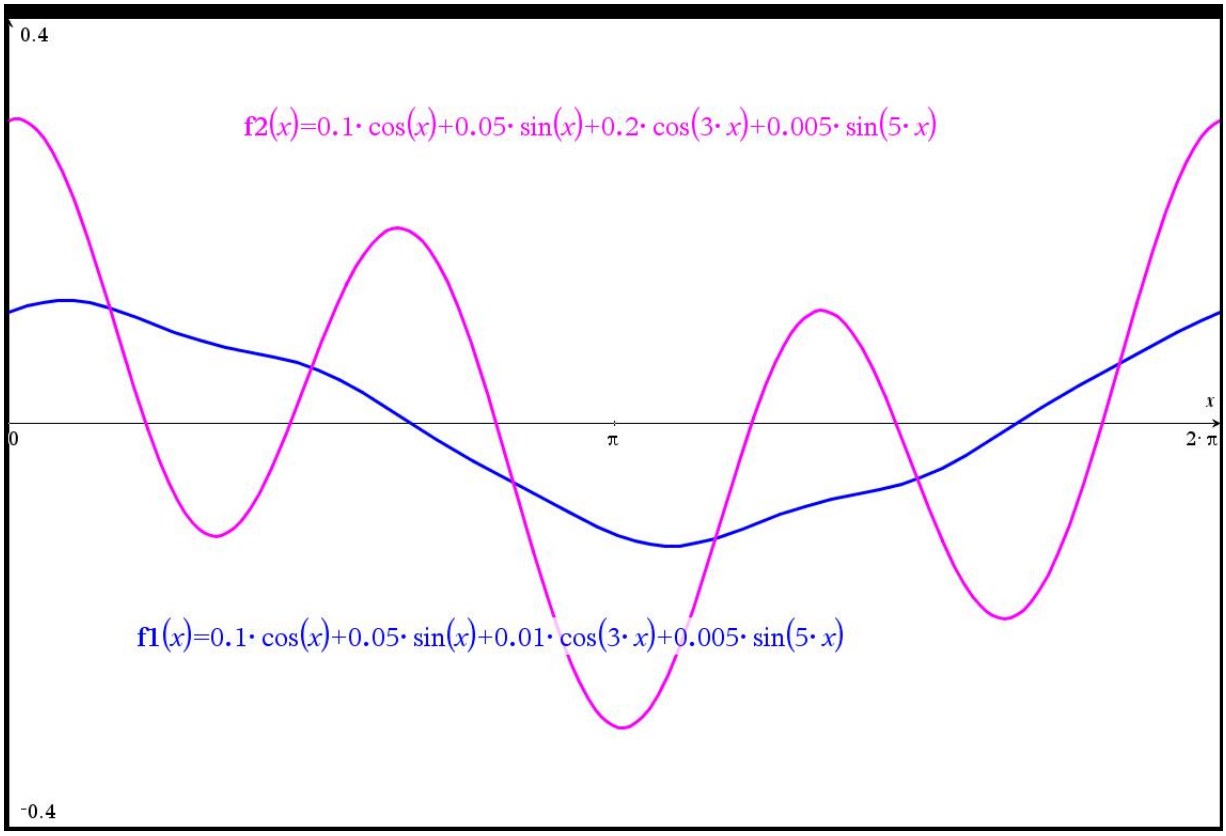
Some facts on trigonometric polynomials: if n is a positive integer, the functions $\sin(nx)$ and $\cos(nx)$ are periodic of period $\frac{2\pi}{n}$, so 2π is also a period.

Example: the following expressions are two $2p$-periodic functions of the variable x:

$$0.1\cos(x)+0.05\sin(x)+\mathbf{0.01}\cos(3x)+0.005\sin(4x)$$

$$0.1\cos(x)+0.05\sin(x)+\mathbf{0.2}\cos(3x)+0.005\sin(4x)$$

Thinking of a mass-spring problem, the difference is in the frequency with which the mass oscillates back and forth.



2.1 Use of Fourier series

Here is a procedure for solving the ODE

$$m\,y''(t) + b\,y'(t) + k\,y(t) = f(t)$$

where m, b, and k are *positive* constants and where $f(t)$ is a *nontrivial periodic input.*

We replace $f(t)$ by its Fourier series. For simplicity, we can suppose the coefficient a_0 of $f(t)$ to be 0 (if not, the constant a_0/k is added to the steady-state solution): so we have

$$f(t) \sim \sum_{n=1}^{\infty} \left(a_n \cos(nt) + b_n \sin(nt) \right)$$

(a word on convergence)

The method of undetermined coefficients is applied in order to find a particular solution of the ODE

$$m\,y''(t) + b\,y'(t) + k\,y(t) = a_n \cos(nt) + b_n \sin(nt)$$

Because $b > 0$, the candidate for this particular solution is of the form

$$A_n \cos(nt) + B_n \sin(nt)$$

the amplitude being $\sqrt{A_n^2 + B_n^2}$.

Because of the linearity of the ODE, the principle of superposition applies and the steady-state solution will also be a 2π-periodic function represented by the Fourier series

$$\sum_{n=1}^{\infty} \left(A_n \cos(nt) + B_n \sin(nt) \right)$$

Exactly as we did before, we can find this:

$$\sqrt{A_n^2+B_n^2}=\frac{\sqrt{a_n^2+b_n^2}}{\sqrt{b^2n^2+\left(k-mn^2\right)^2}}$$

We know that $\lim_{n\to\infty}\sqrt{a_n^2+b_n^2}=0.$

$$\sqrt{A_n^2+B_n^2}=\frac{\sqrt{a_n^2+b_n^2}}{\sqrt{b^2n^2+\left(k-mn^2\right)^2}}$$

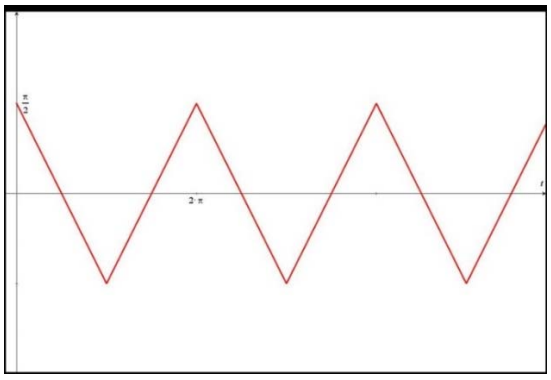
So, we also have $\lim_{n\to\infty}\sqrt{A_n^2+B_n^2}=0.$

But we must be careful if $b^2 < 2mk$: in some examples, the amplitude of the terms of order n close to

$$n\approx\sqrt{\frac{k}{m}-\frac{b^2}{2m^2}}$$

can be large enough and **an almost harmonic oscillation output with frequency *n* times the frequency of the input will result.**

An example: let $f(t)$ be the former input:



This is the 2π-periodic function defined by

$$f(t)=\begin{cases}\pi/2-t, & 0<t<\pi\\ t-3\pi/2, & \pi<t<2\pi\end{cases}\qquad f(t+2\pi)=f(t)$$

We will find the steady-state solution to each of these ODEs:

$$y''(t)+3\,y'(t)+2\,y(t)=f(t)$$
$$y''(t)+0.1\,y'(t)+9\,y(t)=f(t)$$

Do we have $b^2 < 2mk$? Note that $3^2 = 9 > 2\cdot1\cdot2 = 4$ But $0.1^2 = 0.01 < 2\cdot1\cdot9 = 18$.

Then compute $\sqrt{\frac{k}{m}-\frac{b^2}{2m^2}}\to\sqrt{\frac{9}{1}-\frac{0.1^2}{2\cdot1^2}}\approx2.99917$

Using Nspire CX CAS, we found the following results:

For $y''(t)+3\,y'(t)+2\,y(t)=f(t)$, the steady-state solution is well approximated by

$$\frac{2\cos(t)}{5\pi}+\frac{6\sin(t)}{5\pi}\approx0.402634\sin(t+0.321751)$$

This is because the sequence of amplitudes is

{**0.402634**, 0, 0.012408, 0, 0.001855, 0, 0.000505, 0, 0.000188, …..}

While for $y''(t) + 0.1\, y'(t) + 9\, y(t) = f(t)$, the steady-state solution is well approximated by

$$\frac{3200\cos(t)}{6401\pi} + \frac{40\sin(t)}{6401\pi} + \frac{40\sin(3t)}{27\pi} \approx$$
$$0.159143\sin(t + 1.5583) + 0.47157\sin(3t)$$

This is because the sequence of amplitudes is

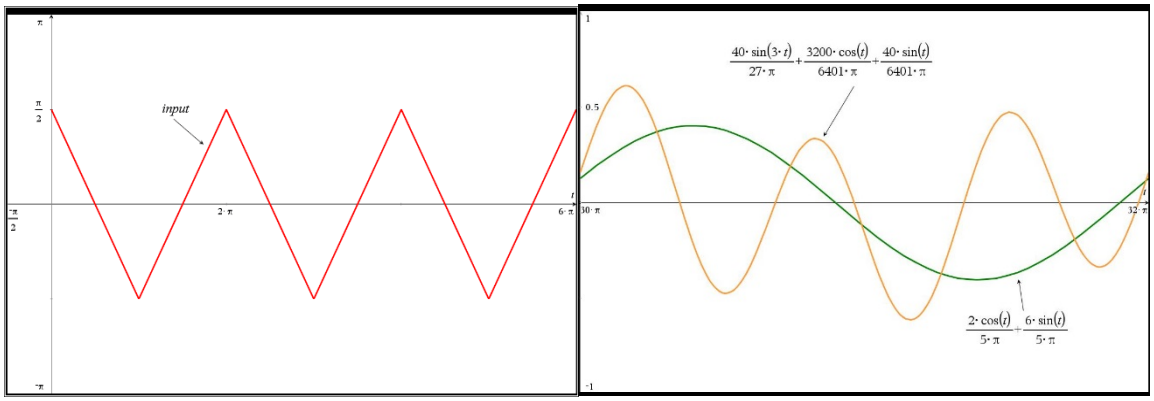
{0.159143, 0, **0.47157**, 0, 0.003182, 0,0.00065, 0, 0.000218, …….}

Let us summarize in the next slide.

A (satisfactory) steady-state output of the ODE

$$y''(t) + 3\, y'(t) + 2\, y(t) = input$$
$$(\text{resp. } y''(t) + 0.1\, y'(t) + 9\, y(t) = input)$$

is the **green** (resp. **orange**) curve (shown on the interval $30\pi < t < 32\pi$).



Let us move again to Nspire and see how we found the last results.

Suppose we have the ODE

$m\cdot y''(t) + b\cdot y'(t) + k\cdot y(t) = f(t)$

where the external force $f(t)$ is the $2\cdot\pi$−periodic function

$$\mathbf{f1}(t) = \begin{cases} \frac{\pi}{2} - t, & 0 < t < \pi \\ t - \frac{3\cdot\pi}{2}, & \pi < t < 2\cdot\pi \end{cases} \qquad \mathbf{f1}(t + 2\cdot\pi) = \mathbf{f1}(t)$$

The fourier partial sum or order 9 of **f1** is (note: **f1** is an even function)

$$\text{fourier}(\mathbf{f1}(t), t, 0, 2\cdot\pi, 9) \blacktriangleright \frac{4\cdot\cos(9\cdot t)}{81\cdot\pi} + \frac{4\cdot\cos(7\cdot t)}{49\cdot\pi} + \frac{4\cdot\cos(5\cdot t)}{25\cdot\pi} + \frac{4\cdot\cos(3\cdot t)}{9\cdot\pi} + \frac{4\cdot\cos(t)}{\pi}$$

So the Fourier coefficients of **f1** are: $a_0 = 0$, $a_n = \frac{2\cdot\left(1-(-1)^n\right)}{n^2\cdot\pi}$, $b_n = 0$

The harmonic of order n of the steady−state solution has the amplitude

$$\frac{\sqrt{a_n{}^2 + b_n{}^2}}{\sqrt{b^2\cdot n^2 + \left(k - m\cdot n^2\right)^2}}$$

$$\mathbf{de1}(y):=\frac{d^2}{dt^2}(y)+3\cdot\frac{d}{dt}(y)+2\cdot y \;\blacktriangleright\; \textit{Done}$$

$$\mathbf{de2}(y):=\frac{d^2}{dt^2}(y)+1/10\cdot\frac{d}{dt}(y)+9\cdot y \;\blacktriangleright\; \textit{Done}$$

$$\mathbf{ex1}(n):=\frac{\sqrt{\left(\frac{2\cdot\left(1-(-1)^n\right)}{n^2\cdot\pi}\right)^2+0}}{\sqrt{3^2\cdot n^2+\left(2-1\cdot n^2\right)^2}} \;\blacktriangleright\; \textit{Done}$$

$$\text{approx}\left(\text{seq}\left(\mathbf{ex1}(n),n,1,9\right)\right) \;\blacktriangleright\; \{0.402634,0.,0.012408,0.,0.001855,0.,0.000505,0.,0.000188\}$$

$$\mathbf{ex2}(n):=\frac{\sqrt{\left(\frac{2\cdot\left(1-(-1)^n\right)}{n^2\cdot\pi}\right)^2+0}}{\sqrt{(0.1)^2\cdot n^2+\left(9-1\cdot n^2\right)^2}} \;\blacktriangleright\; \textit{Done}$$

$$\text{approx}\left(\text{seq}\left(\mathbf{ex2}(n),n,1,9\right)\right) \;\blacktriangleright\; \{0.159143,0.,0.47157,0.,0.003182,0.,0.00065,0.,0.000218\}$$

$$\mathbf{can}:=\alpha\cdot\cos(n\cdot t)+\beta\cdot\sin(n\cdot t) \;\blacktriangleright\; \alpha\cdot\cos(n\cdot t)+\beta\cdot\sin(n\cdot t)$$

$$\mathbf{de1}(\mathbf{can}) \;\blacktriangleright\; \left(-n^2\cdot\alpha+3\cdot n\cdot\beta+2\cdot\alpha\right)\cdot\cos(n\cdot t)+\left(-n^2\cdot\beta-3\cdot n\cdot\alpha+2\cdot\beta\right)\cdot\sin(n\cdot t)$$

$$\mathbf{de2}(\mathbf{can}) \;\blacktriangleright\; \left(-n^2\cdot\alpha+\frac{n\cdot\beta}{10}+9\cdot\alpha\right)\cdot\cos(n\cdot t)+\left(-n^2\cdot\beta-\frac{n\cdot\alpha}{10}+9\cdot\beta\right)\cdot\sin(n\cdot t)$$

$$\text{solve}\left(\begin{cases}-n^2\cdot\alpha+3\cdot n\cdot\beta+2\cdot\alpha=\frac{2\cdot\left(1-(-1)^n\right)}{n^2\cdot\pi}\\ -n^2\cdot\beta-3\cdot n\cdot\alpha+2\cdot\beta=0\end{cases},\{\alpha,\beta\}\right)$$

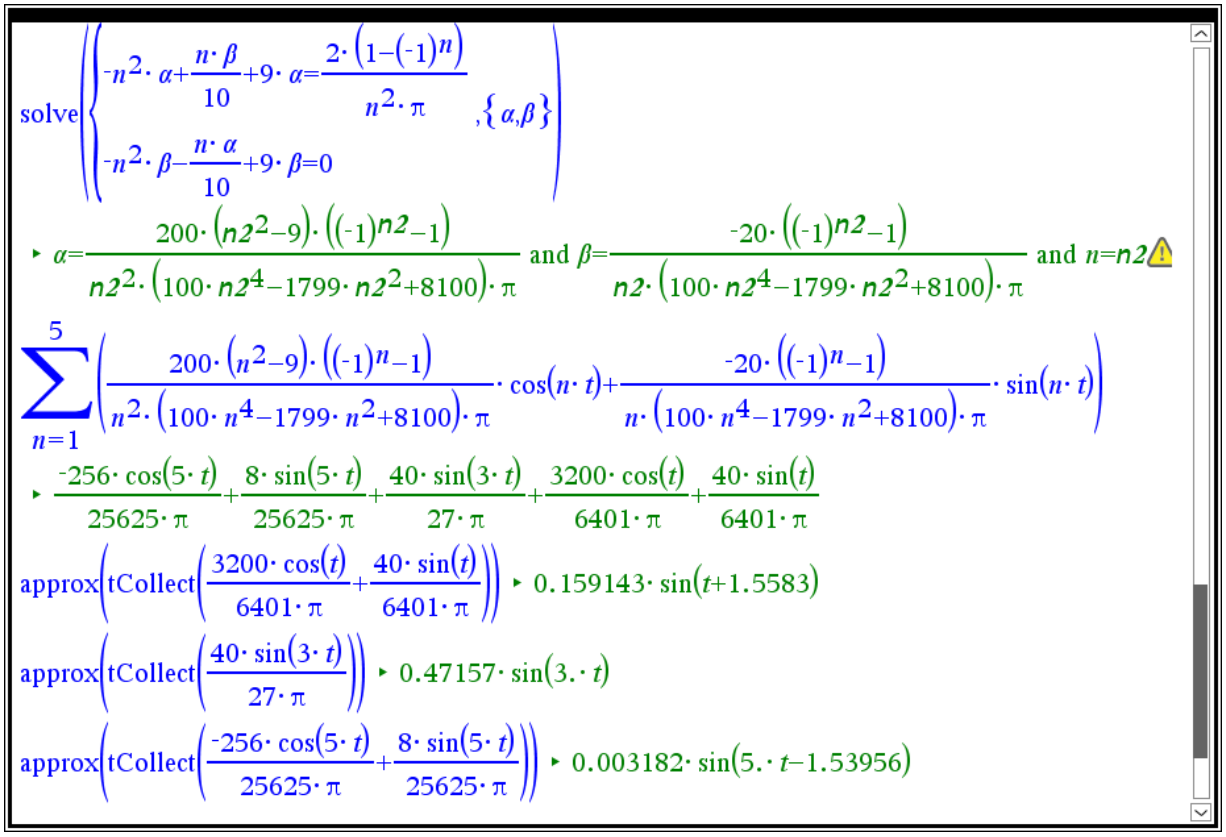
$$\blacktriangleright\; \alpha=\frac{2\cdot\left(n1^2-2\right)\cdot\left((-1)^{n1}-1\right)}{n1^2\cdot\left(n1^4+5\cdot n1^2+4\right)\cdot\pi} \text{ and } \beta=\frac{-6\cdot\left((-1)^{n1}-1\right)}{n1\cdot\left(n1^4+5\cdot n1^2+4\right)\cdot\pi} \text{ and } n=n1$$

$$\sum_{n=1}^{3}\left(\frac{2\cdot\left(n^2-2\right)\cdot\left((-1)^n-1\right)}{n^2\cdot\left(n^4+5\cdot n^2+4\right)\cdot\pi}\cdot\cos(n\cdot t)+\frac{-6\cdot\left((-1)^n-1\right)}{n\cdot\left(n^4+5\cdot n^2+4\right)\cdot\pi}\cdot\sin(n\cdot t)\right)$$

$$\blacktriangleright\; \frac{-14\cdot\cos(3\cdot t)}{585\cdot\pi}+\frac{2\cdot\sin(3\cdot t)}{65\cdot\pi}+\frac{2\cdot\cos(t)}{5\cdot\pi}+\frac{6\cdot\sin(t)}{5\cdot\pi}$$

$$\text{approx}\left(\text{tCollect}\left(\frac{2\cdot\cos(t)}{5\cdot\pi}+\frac{6\cdot\sin(t)}{5\cdot\pi}\right)\right) \;\blacktriangleright\; 0.402634\cdot\sin(t+0.321751)$$

$$\text{approx}\left(\text{tCollect}\left(\frac{-14\cdot\cos(3\cdot t)}{585\cdot\pi}+\frac{2\cdot\sin(3\cdot t)}{65\cdot\pi}\right)\right) \;\blacktriangleright\; 0.012408\cdot\sin(3.\cdot t-0.661043)$$



Here are the last steps of my DERIVE treatment (Beaudin_3.dfw):

#30: SOLUTIONS

#31:

#32:

#33:

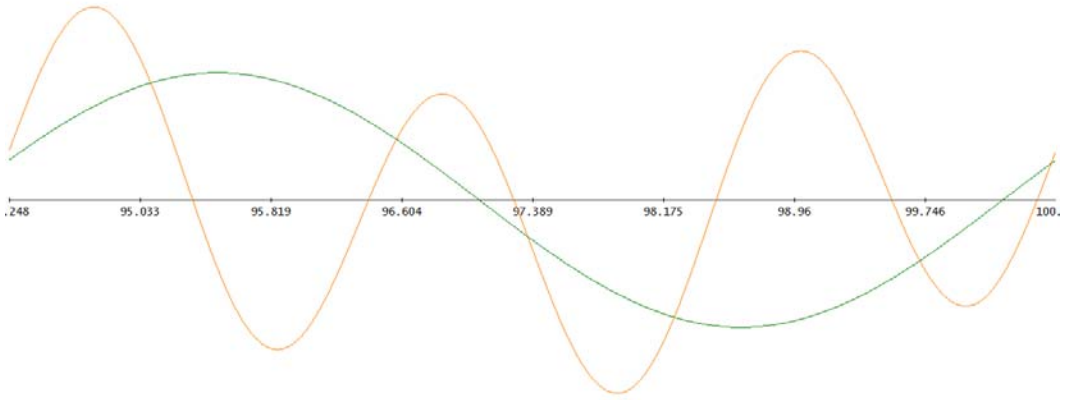
#34: Trigonometry := Collect

#35: APPROX

#36: APPROX

#37: APPROX

Followed by the respective plot (compare with the plot on page 13):



2.2 Validation by convolution

Let us conclude this talk by finding not only the steady-state solution but the *entire exact* solution to each of the last two ODEs using Laplace transforms techniques.

Of course, as t will be increasing, both solutions will be undistinguishable.

Because we can take zero initial conditions, the convolution of the input and the impulse response will be used.

Let say a word about that.

Fact: the solution $y(t)$ to the problem

$$m\,y''(t) + b\,y'(t) + k\,y(t) = f(t),\ \ y(0) = y'(0) = 0$$

is given by the *convolution integral*

$$y(t) = f(t) * h(t) \equiv \int_0^t f(\tau)h(t-\tau)d\tau$$

The function $h(t)$ is the so-called *impulse response*: this is the inverse Laplace transform of the *transfer function*

$$\frac{1}{m\,s^2 + b\,s + k}$$

This is a very useful result … but computing the convolution when one of the functions is a nontrivial periodic function defined using a "modulo" function is quite difficult!

This is why Fourier series stuff is still important. But because Laplace transforms stuff is also part of engineering students' curriculum, why not *try* to validate our results using the convolution of the input with the system impulse response?

A symbolic system might be unable to find a closed-form for the convolution integral. In fact, Nspire CX CAS does not succeed (*Derive* is able to do it but this is a huge expression).

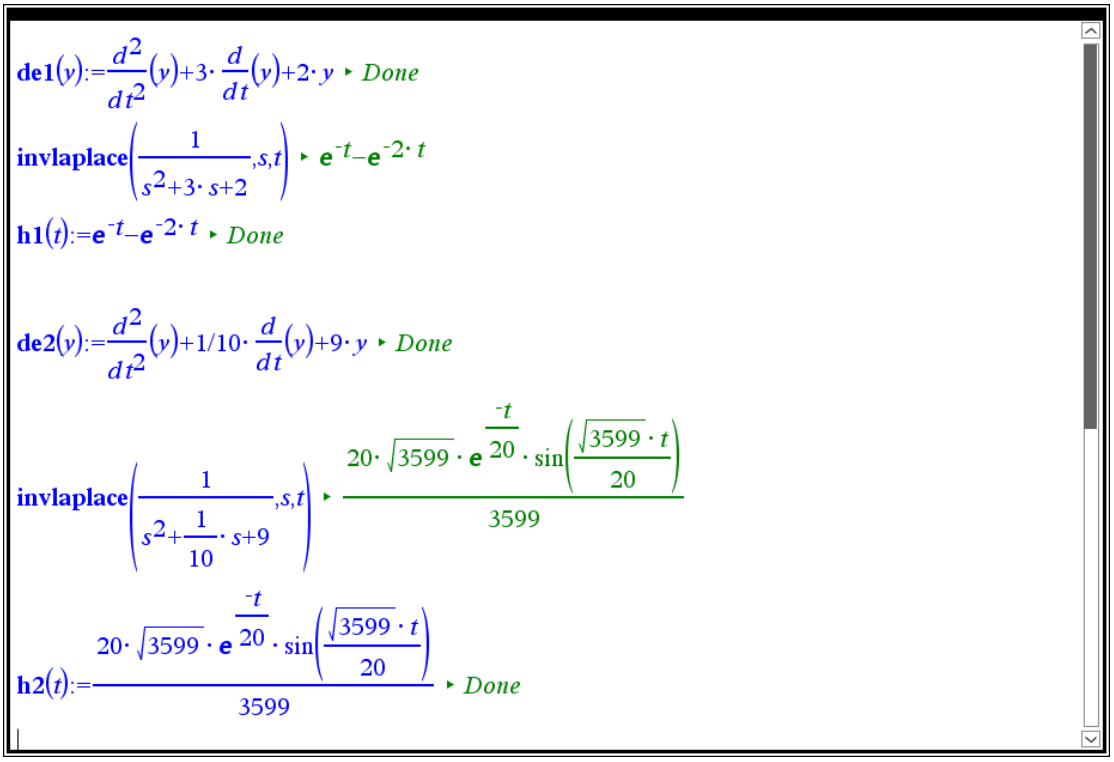
But even if a system is unable to compute the convolution in closed-form, there is always the possibility to plot the graph and this would become a validation of our earlier results.

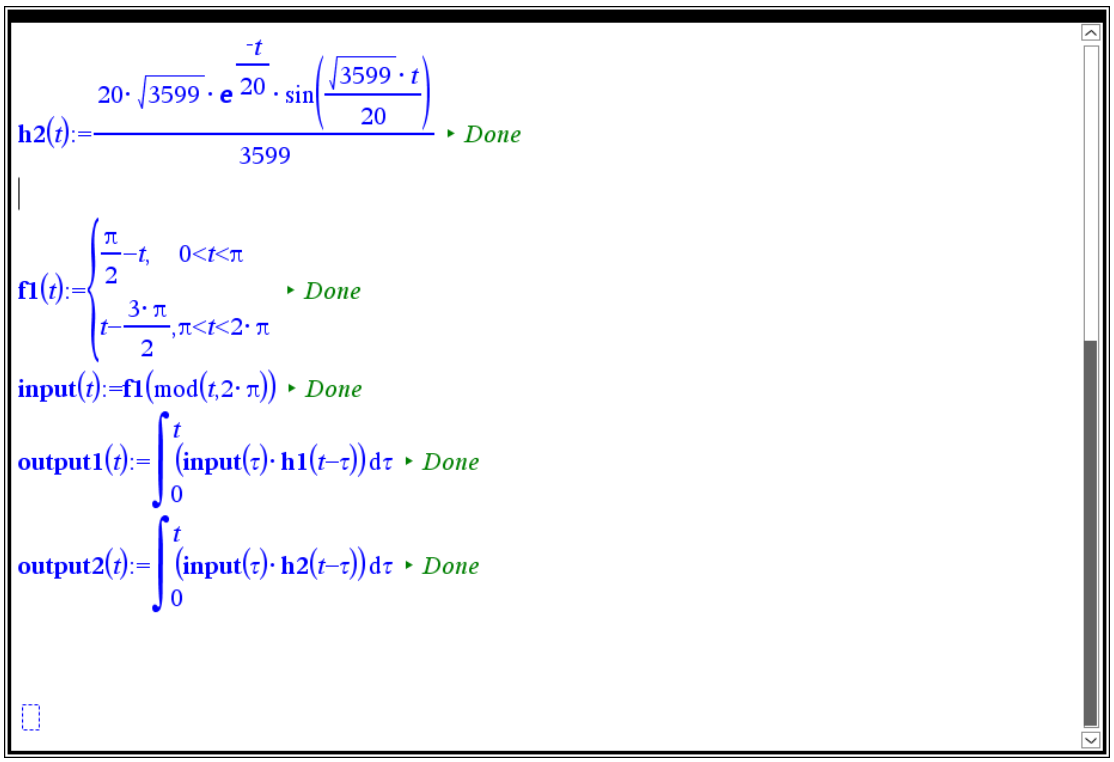
The next slides compare the steady-state solution and the exact solution (assuming zero initial conditions) for each of the ODEs

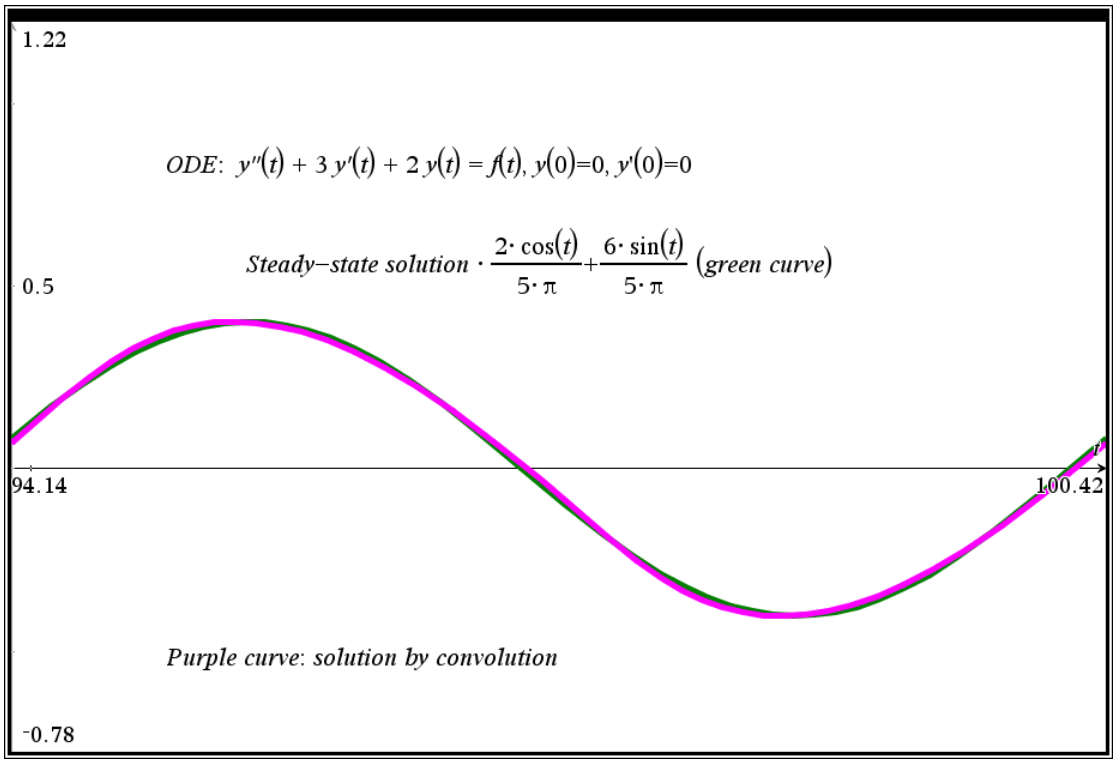
$$y''(t) + 3\,y'(t) + 2\,y(t) = f(t)$$

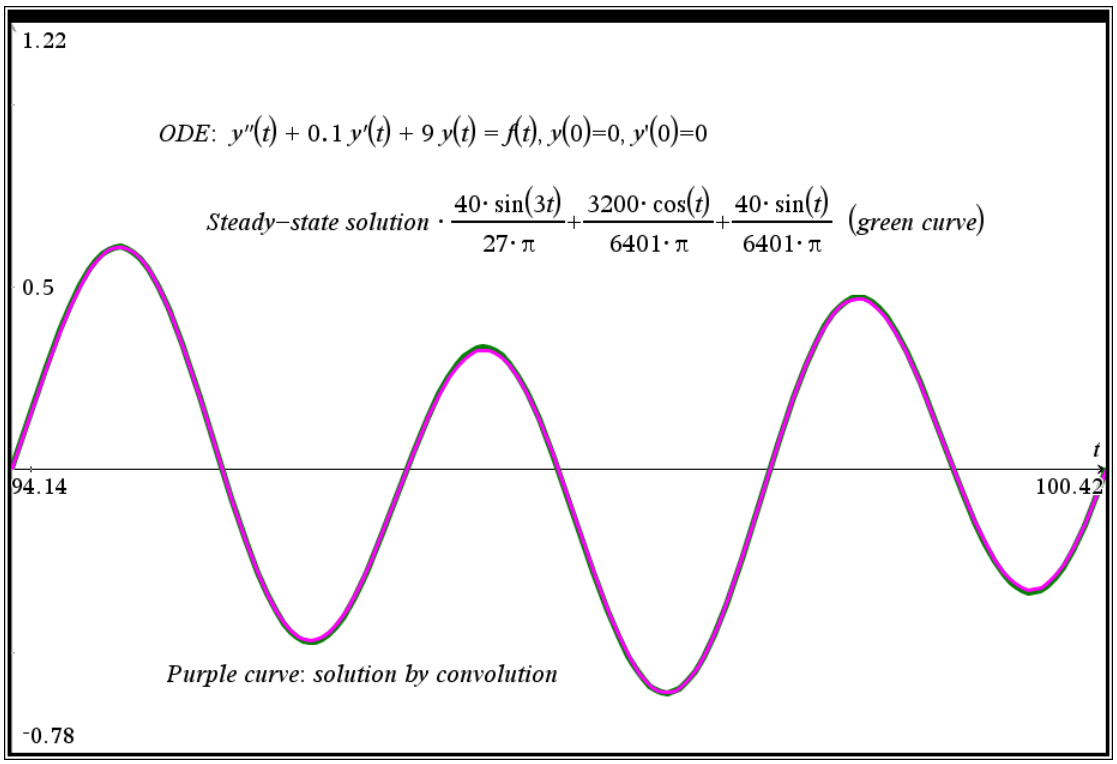
$$y''(t) + 0.1\,y'(t) + 9\,y(t) = f(t)$$

We used again Nspire to plot the graphs and we have chosen the interval $30\pi < t < 32\pi$. Here are the screen images of the graphs.









Thank You!

In Beaudin_4.dfw I perform the most difficult chapter of Michel's presentation.

I start defining the input-function as a piecewise defined function (if-construct).

```
      f1(t) :=
        If 0 < t < π
           π/2 - t
#1:        If π < t < 2·π
              t - 3·π/2
              ?

#2:   input(t) := f1(MOD(t, 2·π))
```

$$\#3: \quad \left[de1(y) := \left(\frac{d}{dt}\right)^2 y + 3\cdot\frac{d}{dt} y + 2\cdot y,\ de2(y) := \left(\frac{d}{dt}\right)^2 y + \frac{1}{10}\cdot\frac{d}{dt} y + 9\cdot y \right]$$

$$\#4: \quad h1(t) := \text{INVLaplace}\left(\frac{1}{s^2 + 3\cdot s + 2},\ s,\ t\right)$$

$$\#5: \quad h1(t) := e^{-t} - e^{-2\cdot t}$$

$$\#6: \quad h2(t) := \text{INVLaplace}\left(\frac{1}{s^2 + 0.1\cdot s + 9},\ s,\ t\right)$$

Terrence Etchells programmed an INVLaplace function for DERIVE (contained in the Users/Transforms-folder) but it is not able to simplify #6, approximating works well (#7).

$$\#7: \quad h2(t) := 0.3333796392\cdot e^{-0.05\cdot t}\cdot \text{SIN}(2.999583304\cdot t)$$

$$\#8: \quad \int_0^t \text{input}(\tau)\cdot h1(t-\tau)\ d\tau$$

$$\#9: \quad \int_0^t \left(e^{\tau - t} - e^{2\cdot\tau - 2\cdot t}\right)\cdot \text{IF}\left(-1146408 < -\text{MOD}(364913\cdot\tau, 2292816) < 0,\ \frac{\pi}{2} - \frac{\text{MOD}(364913\cdot\tau, 2292816)}{364913},\ \text{IF}\left(\pi < \frac{\text{MOD}(364913\cdot\tau, 2292816)}{364913} < 2\cdot\pi,\ \frac{\text{MOD}(364913\cdot\tau, 2292816)}{364913} - \frac{3\cdot\pi}{2},\ ?\right)\right) d\tau$$

This definition of f(t) does not allow integration. So, we don't find a closed solution.

$$\#10: \quad f2(t) := \left(\frac{\pi}{2} - t\right)\cdot\chi(0,\ t,\ \pi) + \left(t - \frac{3\cdot\pi}{2}\right)\cdot\chi(\pi,\ t,\ 2\cdot\pi)$$

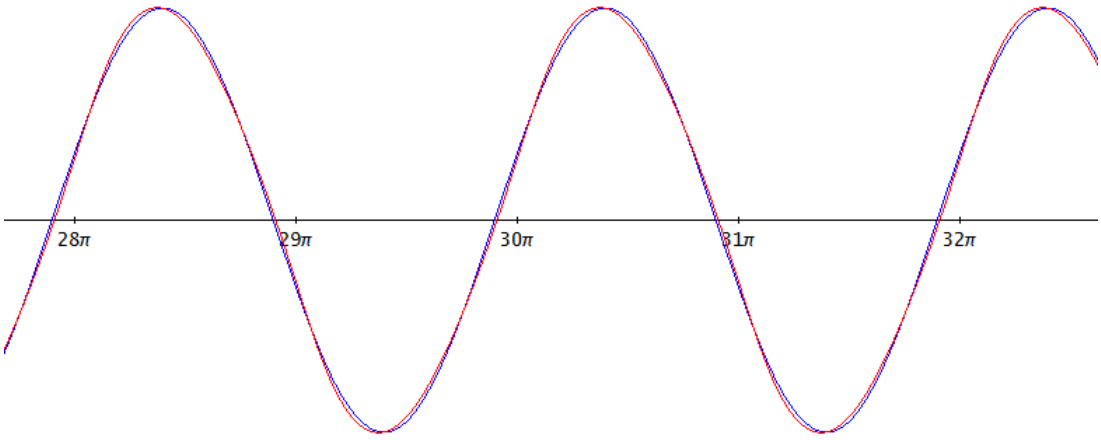
```
#11:  input2(t) := f2(MOD(t, 2·π))
```

$$\#12: \quad \int_0^t \text{input2}(\tau)\cdot h1(t-\tau)\ d\tau$$

$$\#13:\quad \mathrm{SIGN}(2292816 - \mathrm{MOD}(364913\cdot t,\ 2292816))\cdot\left(\frac{\mathrm{MOD}(364913\cdot t,\ 2292816)}{1459652} + \frac{15123393553\cdot e^{4585632/364913 - 2\cdot \mathrm{MOD}(364913\cdot t,\ 2292816)/364913}}{56494006095} - \frac{64493142329\cdot e^{2292816/364913 - \mathrm{MOD}(364913\cdot t,\ 2292816)/364913}}{225976024380} - \frac{961771}{619260}\right) +$$

The other form defining the periodic function gives a closed solution for the integral, which is a huge expression (15 lines). See its plot (red) together with the steady state-solution (blue):

$$\#14:\quad \frac{2\cdot\cos(t)}{5\cdot\pi} + \frac{6\cdot\sin(t)}{5\cdot\pi}$$



It is interesting that expression #9 can be plotted successfully and gives the same red curve. We perform the same procedure with the approximated inverse Laplace of the second input-function:

$$\#15:\quad \int_0^t \mathrm{input2}(\tau)\cdot h2(t-\tau)\,d\tau$$

$$\#16:\quad \mathrm{SIGN}(2292816 - \mathrm{MOD}(364913\cdot t,\ 2292816))\cdot\left(\frac{41466240200\cdot \mathrm{MOD}(364913\cdot t,\ 2292816)}{272368261982211713} + e^{-}\right.$$

The integral shows 55 lines (more than 40 lines when approximated).

$$\#18:\quad \frac{40\cdot\sin(3\cdot t)}{27\cdot\pi} + \frac{3200\cdot\cos(t)}{6401\cdot\pi} + \frac{40\cdot\sin(t)}{6401\cdot\pi}$$

The plots will follow – I have the ambition to find the inv. Laplacian of the 2nd input-function.

#19: $\text{SOLUTIONS}(s^2 + 0.1\cdot s + 9 = 0,\ s) = \left[-\frac{1}{20} + \frac{\sqrt{3599}\cdot i}{20},\ -\frac{1}{20} - \frac{\sqrt{3599}\cdot i}{20}\right]$

#20: $\left[\alpha := -\frac{1}{20} + \frac{\sqrt{3599}\cdot i}{20},\ \beta := -\frac{1}{20} - \frac{\sqrt{3599}\cdot i}{20}\right]$

#21: $\frac{1}{s^2 + \frac{s}{10} + 9} = \frac{a}{s-\alpha} + \frac{b}{s-\beta}$

#22: $1 = a\cdot(s-\beta) + b\cdot(s-\alpha)$

#23: $1 = s\cdot(a+b) + \frac{a+b}{20} + i\cdot\left(\frac{\sqrt{3599}\cdot a}{20} - \frac{\sqrt{3599}\cdot b}{20}\right)$

#24: $\text{SOLUTIONS}\left(\left[a+b=0,\ \frac{a+b}{20} + i\cdot\left(\frac{\sqrt{3599}\cdot b}{20} - \frac{\sqrt{3599}\cdot a}{20}\right) = 1\right],\ [a,\ b]\right)$

#25: $\left[\left[\frac{10\cdot\sqrt{3599}\cdot i}{3599},\ -\frac{10\cdot\sqrt{3599}\cdot i}{3599}\right]\right]$

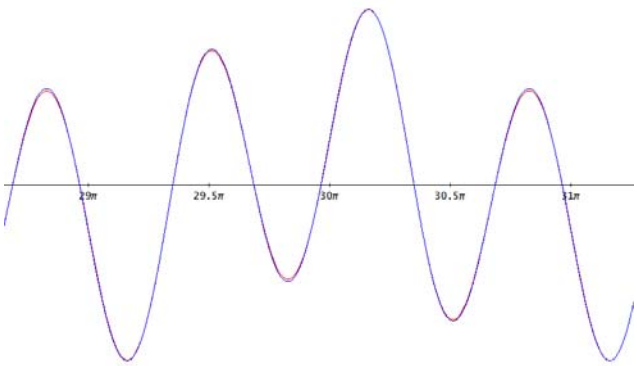
#26: $h2\_(t) := \frac{10\cdot\sqrt{3599}\cdot i}{3599}\cdot\text{EXP}(-\alpha\cdot t) - \frac{10\cdot\sqrt{3599}\cdot i}{3599}\cdot\text{EXP}(-\beta\cdot t)$

#27: $h2\_(t) := \frac{20\cdot\sqrt{3599}\cdot e^{t/20}\cdot\text{SIN}\left(\frac{\sqrt{3599}\cdot t}{20}\right)}{3599}$

Lucky me, it is identical with *h2(t)* on page 17. This was much easier than “translating” Ph. Fortin’s and L. Fredericksen’s sophisticated TI-Nspire-function into “DERIVIAN” (contained in ETS_specfunc.tns).

#28: $\int_0^t \text{input2}(\tau)\cdot h2\_(t-\tau)\ d\tau$

#29: $\text{SIGN}(2292816 - \text{MOD}(364913\cdot t,\ 2292816))\cdot\left(\frac{\text{MOD}(364913\cdot t,\ 2292816)}{6568434} + e^{-\text{MOD}(364913\cdot t,}\right.$

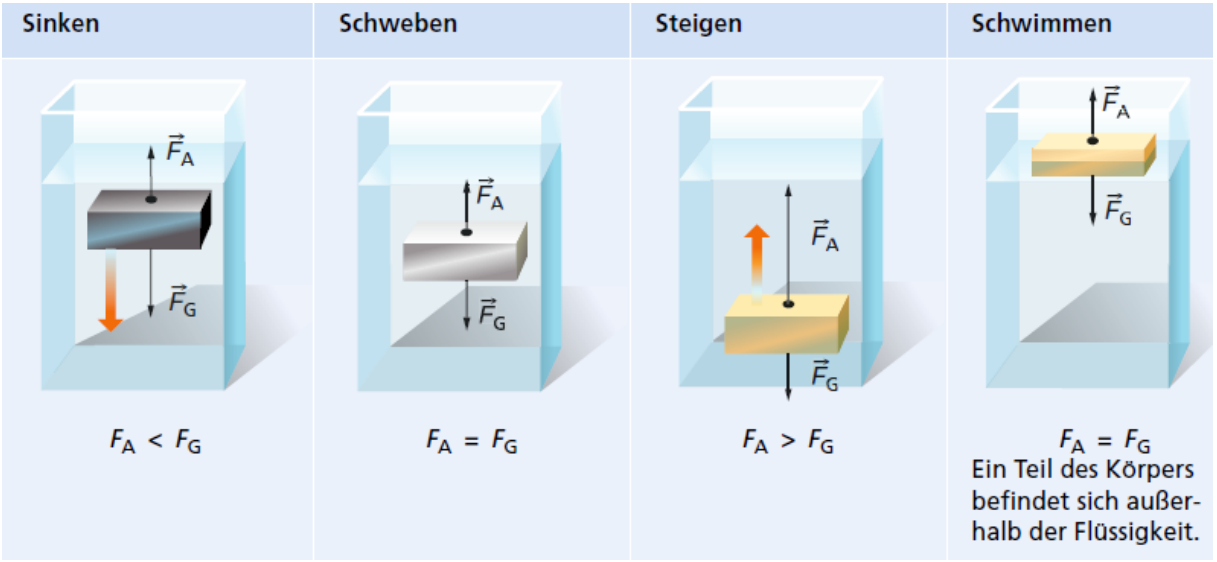


Again, convolution in red and steady state-solution in blue.

Warum schwimmt ein Schiff?

Wolfgang Alvermann, Hinte, Deutschland

Die Frage eines Studierenden der Fachschule-Technik mit Schwerpunkt Schiffbau ist einfach zu beantworten; das spezifische Gewicht eines Körpers (Schiffes) muss kleiner sein als das spezifische Gewicht der Flüssigkeit, in der es schwimmt. Man kann folgende Fälle unterscheiden:



(Grafik aus Duden-Paetec: Physik Gesamtband Sek.I, S. 211)

Am Beispiel einer Hohlkugel aus Metall mit gegebenem Innendurchmesser d_i soll bestimmt werden, bis zu welchem Außendurchmesser d_a diese in der Flüssigkeit schwimmt, bei welchem Außendurchmesser sie schwebt, und wann sie sinkt.

Für den Fall des Schwimmens soll außerdem die Eintauchtiefe ermittelt werden.

Bezeichnungen – Formeln – bestimmte Zahlenwerte

Volumen einer Kugel: $V_K = \frac{\pi}{6} \cdot d_a^3$

Volumen einer Hohlkugel: $V_{Hk} = \frac{\pi}{6} \cdot \left(d_a^3 - d_i^3 \right)$

Dichte einer Flüssigkeit: ρ_{Fl} für Wasser gilt: $\rho_{Fl} = 1 \frac{kg}{dm^3}$

Dichte eines Werkstoffs: ρ_W für Stahl gilt: $\rho_W = 7.85 \frac{kg}{dm^3}$

Dichte eines Gases: ρ_G für Luft gilt: $\rho_G = 1.293 \frac{kg}{m^3}$

Masse: $m = V \cdot \rho$

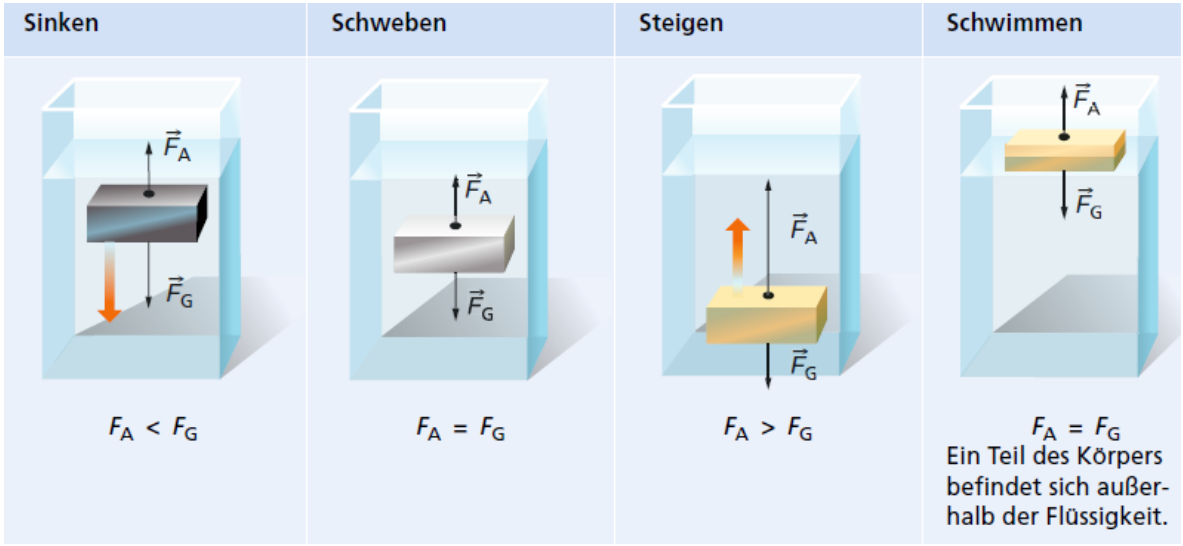
Einheiten beachten **(ρ_G)**

Why does a Ship Swim?

Wolfgang Alvermann, Hinte, Germany & J. Böhm, Austria

The question of a student of a technical college with focus on ship building can be answered easily; the specific gravity of a body (a ship) must be less than the specific gravity of the liquid in which it is swimming. We can distinguish the following cases:

Sinking | Floating | Rising | Swimming



Graph from Duden-Paetec: Physik Gesamtband Sek I, p 211

One part of the body is outside of the liquid

We take a hollow metal sphere made from metal with given inner diameter d_i and we want to determine until which outer diameter d_o the sphere will swim, float, and when it will start sinking.

In case of swimming we will calculate the depth of immersion.

Descriptions – Formulae – Given numerical Values

Volume of a sphere with diameter *d*: $vs(d) := \frac{\pi}{6} \cdot d^3$ ▸ *Done*

Volume of a hollow sphere with diameters *do* and *di*: $vhs(do,di) := \frac{\pi}{6} \cdot (do^3 - di^3)$ ▸ *Done*

We make use of the built-in unit conversion tool!

Density of a liquid *ρl* – for water: $\rho l := 1 \cdot \frac{\_kg}{\_dm^3}$ ▸ $1000. \cdot \frac{\_kg}{\_m^3}$

Density of a material *ρm* – for steel: $\rho m := 7.85 \cdot \frac{\_kg}{\_dm^3}$ ▸ $7850. \cdot \frac{\_kg}{\_m^3}$

Density of a gas *ρg* – for air: $\rho g := 1.293 \cdot \frac{\_kg}{\_m^3}$ ▸ $1.293 \cdot \frac{\_kg}{\_m^3}$

Mass = Volume * Density: $mass(v,\rho) := v \cdot \rho$ ▸ *Done*

Take care for the units (ρG)! - TI-Nspire does the job for us!

Berechnungen: Gegeben ist d_i

Masse der eingeschlossenen Luft: $m_L = \frac{\pi}{6} \cdot d_i^3 \cdot \rho_G \approx 0.000677 \cdot d_i^3$

Masse der Hohlkugel: $m_{Hk} = \frac{\pi}{6} \cdot \rho_W \cdot \left(d_a^3 - d_i^3\right) \approx 4.11025 \cdot \left(d_a^3 - d_i^3\right)$

Gesamtmasse: $m_{ges} = 4.11025 \cdot d_a^3 - 4.10957 \cdot d_i^3$

Nun lässt sich die Dichte der Hohlkugel aus der Gesamtmasse und dem äußeren Kugelvolumen berechnen:

$$\rho_{Hk} = \frac{m_{ges}}{V} = \frac{4.11025 \cdot d_a^3 - 4.10957 \cdot d_i^3}{\frac{\pi}{6} \cdot d_a^3} = 7.85 - \frac{7.848707 \cdot d_i^3}{d_a^3}$$

Es wird $\rho_{Hk} = 1$ gesetzt, um den Schwebezustand zu bestimmen, daraus lassen sich die anderen Fälle ableiten.

Aus $7.85 - \frac{7.848707 \cdot d_i^3}{d_a^3} = 1$ folgt: $\boxed{d_{as} \approx 1.0464 \cdot d_i}$ der max. Außendurchmesser

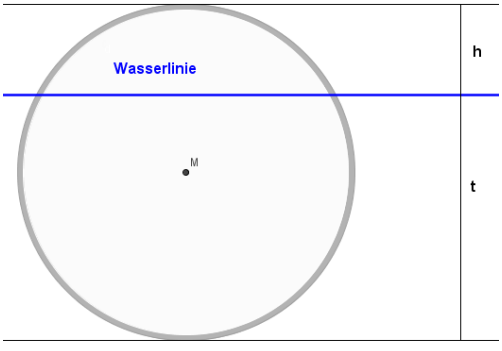
Wählt man für $d_i = 1m = 10\,dm$, ergibt sich folgendes:

d_a < 10.464 dm → Hohlkugel schwimmt

d_a = 10.464 dm → Hohlkugel schwebt

d_a > 10.464 dm → Hohlkugel sinkt

Für d_a = 10.4 dm, also einer Wandstärke von 2 cm, ergibt sich eine Gesamtmasse von $m_{Hk} \approx 513.9\,kg$, ein Volumen von $V = 588.977\,dm^3$, so dass man ein spezifisches Gewicht (Dichte) von $\rho_{Hk} \approx 0.87253\,\frac{kg}{dm^3}$ erhält → der Hohlkörper schwimmt also.



Nun soll die Eintauchtiefe dieses Hohlkörpers mit den gegebenen Maßen berechnet werden.

Es gilt: $t = d_a - h$

Für die Gewichtskraft der Hohlkugel erhält man

$F_{Hk} = 5041.35\,N$

Calculations: Given is d_i

Mass of the enclosed air: $ma:=mass(vs(di),\rho g) \rightarrow 0.677013\cdot di^3\cdot \frac{\_kg}{\_m^3}$

Mass of the hollow sphere: $mhs:=mass(vhs(do,di),\rho m) \rightarrow -4110.25\cdot (di^3-do^3)\cdot \frac{\_kg}{\_m^3}$

Total mass: $mtot:=ma+mhs \rightarrow -4109.57\cdot (di^3-1.00016\cdot do^3)\cdot \frac{\_kg}{\_m^3}$

We can calculate the density of the hollow sphere from the total mass and the volume of the outer sphere:

$phs:=zeros(mtot-vs(do)\cdot \rho hs,\rho hs)[1] \rightarrow \frac{-7848.71\cdot (di^3-1.00016\cdot do^3)}{do^3}\cdot \frac{\_kg}{\_m^3}$

We set *phs*=1 (water) in order to find the floating condition; then we deduce the other cases (di=10 dm):

$zeros\left(\frac{1\cdot \_kg}{\_dm^3}-phs,do\right) \rightarrow \{1.04641\cdot di\}$ = maximum outer diameter

$do\_max:=1.04641\cdot di|di=10\cdot \_dm \rightarrow 1.04641\cdot \_m$

$1.04641\cdot di|di=10\cdot \_dm \rightarrow 1.04641\cdot \_m$

Choosing d_i = 1 *m* = 10 *dm*

d_o < 10.464 dm → hollow sphere swims

d_o = 10.464 dm → hollow sphere floats

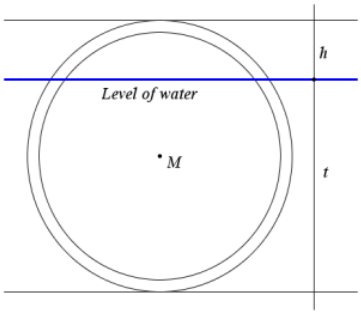
d_o > 10.464 dm → hollow sphere sinks

Let *do*=10.4 dm, i.e. a thickness of 2 *cm* then:

mass = $mtot|di=10\cdot \_dm \text{ and } do=10.4\cdot \_dm \rightarrow 513.899\cdot \_kg$ and

volume = $vs(10.4\cdot \_dm) \rightarrow 0.588977\cdot \_m^3$, then the specific weight (density) of the sphere is

$\rho_{hs} = \frac{513.899\cdot \_kg}{0.588977\cdot \_m^3} \rightarrow 872.528\cdot \frac{\_kg}{\_m^3}$ (≈ 0.872 g/cm³) → the hollow sphere is swimming.



We will calculate the depth of immersion *t* of this hollow body using the given measures.

Let *t* = *do* - *h*

Nachdem die Kugel bzw. die Hohlkugel sich im Wasser im Ruhezustand befindet, gilt:

Gewichtskraft der Kugel = Auftriebskraft.

$5041.35 = \rho_{Fl} \cdot g \cdot V_u$, wobei V_u das Volumen des unter Wasser liegenden Teils der Kugel ist.

$$V_u = \underbrace{\frac{\pi}{6} \cdot d_a^3}_{Kugel} - \underbrace{\pi \cdot h^2 \cdot \left(\frac{d_a}{2} - \frac{h}{3} \right)}_{Kugelabschnitt}$$

Daraus lässt sich die Höhe des herausragenden Teils der Kugel und damit die Eintauchtiefe berechnen.

$$5041.35 = 9.81 \cdot 1 \left[\frac{\pi}{6} \cdot 10.4^3 - \pi \cdot h^2 \cdot \left(\frac{10.4}{2} - \frac{h}{3} \right) \right] \quad \rightarrow \quad \boxed{\begin{array}{l} h = 2.324\ dm \\ t = 8.076\ dm \end{array}}$$

Verallgemeinerungen

Für eine Hohlkugel sollen nun abschließend die bisherigen Ergebnisse verallgemeinert werden.

Gesamtmasse

Für die Masse des eingeschlossenen Gases gilt: $m_g = \frac{\pi}{6 \cdot 1000} \cdot d_i^3 \cdot \rho_g$

Durch den Faktor 1000 werden die Einheiten von ρ_G und ρ_W kompatibel.

Für die Masse der Hohlkugel gilt: $m_{Hk} = \frac{\pi}{6} \cdot \left(d_a^3 - d_i^3 \right) \cdot \rho_W$

$$\boxed{m_{ges} = \frac{\pi}{6} \cdot \left[d_a^3 \cdot \rho_w - \frac{d_i^3}{1000} \cdot \left(1000 \cdot \rho_w - \rho_G \right) \right]}$$

Mit $V = \frac{\pi}{6} \cdot d_a^3$ ermittelt man das spezifische Gewicht der Hohlkugel zu

$$\boxed{\rho_{Hk} = \rho_W - \left(\frac{d_i}{d_a} \right)^3 \cdot \left(\rho_W - \frac{\rho_G}{1000} \right)}$$

Mit dem Ansatz $\rho_{Hk} = 1$ ergibt d_{as} für den Schwebezustand zu

$$\boxed{d_{as} = \frac{d_i}{10 \cdot \sqrt[3]{\frac{\rho_W - 1}{1000 \cdot \rho_W - \rho_G}}}}$$

The weight of the hollow sphere is $\mathbf{fhs} := 513.899 \cdot \_kg \cdot \_g \blacktriangleright 5039.63 \cdot \_N$

As the sphere – better the hollow sphere - is at rest we have:

Weight of the Body = Lifting Force (*).

Let *vu* = volume of the underwater part of the sphere = $V_{Sphere} - V_{Segment}$.

$$\mathbf{vu} := \mathbf{vs}(10.4 \cdot \_dm) - \pi \cdot (h \cdot \_dm)^2 \cdot \left(\frac{d}{2} - \frac{h \cdot \_dm}{3}\right) \Big| d = 10.4 \cdot \_dm$$

$$\blacktriangleright 0.001047 \cdot \left(h^3 - 15.6 \cdot h^2 + 562.432\right) \cdot \_m^3$$

We solve (*) for *h*:

$\mathrm{solve}(\mathbf{fhs} = \_g \cdot \boldsymbol{\rho l} \cdot \mathbf{vu}, h) \blacktriangleright h = -2.01731$ or $h = 2.32384$ or $h = 15.2935$ → h = 2.324 *dm*

Finally: Depth of immersion = $10.4 \cdot dm - 2.32384 \cdot dm \blacktriangleright 8.07616 \cdot dm$

Generalizations

Finally, we will generalize our results we have obtained for the hollow sphere.

Generalizations

$\mathbf{vs}(d) := \frac{\pi}{6} \cdot d^3$ ▸ *Done* $\mathbf{mg}(di, \rho g) := \mathbf{vs}(di) \cdot \rho g$ ▸ *Done* $\mathbf{mg}(di, \rho g) \blacktriangleright \frac{di^3 \cdot \rho g \cdot \pi}{6}$

$\mathbf{mhs}(do, di, \rho m) := (\mathbf{vs}(do) - \mathbf{vs}(di)) \cdot \rho m$ ▸ *Done* $\mathbf{mhs}(do, di, \rho m) \blacktriangleright \frac{-\left(di^3 - do^3\right) \cdot \rho m \cdot \pi}{6}$

$\mathbf{mtot}(do, di, \rho g, \rho m) := \mathbf{mg}(di, \rho g) + \mathbf{mhs}(do, di, \rho m)$ ▸ *Done*

$\mathbf{mtot}(do, di, \rho g, \rho m) \blacktriangleright \frac{\left(di^3 \cdot (\rho g - \rho m) + do^3 \cdot \rho m\right) \cdot \pi}{6}$ → mass of enclosed gas + hollow sphere

$\boldsymbol{\rho}\mathbf{hs}(do, di, \rho g, \rho m) := \frac{\mathbf{mtot}(do, di, \rho g, \rho m)}{\mathbf{vs}(do)}$ ▸ *Done* $\boldsymbol{\rho}\mathbf{hs}(do, di, \rho g, \rho m) \blacktriangleright \frac{di^3 \cdot (\rho g - \rho m) + do^3 \cdot \rho m}{do^3}$

$\mathbf{do\_max}(di, \rho g, \rho m, \rho l) := \mathrm{zeros}(\rho l - \boldsymbol{\rho}\mathbf{hs}(do, di, \rho g, \rho m), do)[1]$ ▸ *Done*

$$\mathbf{do\_max}(di, \rho g, \rho m, \rho l) \blacktriangleright \frac{di \cdot (\rho g - \rho m)^{\frac{1}{3}}}{(\rho l - \rho m)^{\frac{1}{3}}}$$ → outer diameter for floating condition

$$\mathbf{do\_max}\left(10 \cdot \_dm, \frac{1.293 \cdot \_kg}{\_m^3}, \frac{7.85 \cdot \_kg}{\_dm^3}, \frac{1 \cdot \_kg}{\_dm^3}\right) \blacktriangleright 1.04641 \cdot \_m$$

Für die Eintauchtiefe kann man den Ansatz $F_A = F_G$ (siehe Grafik auf Seite 22) verwenden:

$F_A = \rho_{FL} \cdot V_U \cdot g = \rho_{FL} \cdot (V - V_O) \cdot g$; V_O = Kugelabschnitt oberhalb der Wasseroberfläche

$F_G = \rho_{Hk} \cdot V \cdot g$

Daraus folgt:

$$\left.\begin{aligned}
\rho_{FL} \cdot (V - V_O) \cdot \cancel{g} &= \rho_{Hk} \cdot V \cdot \cancel{g} \\
\frac{V - V_O}{V} &= \frac{\rho_{Hk}}{\rho_{Fl}} \\
1 - \frac{V_O}{V} &= \frac{\rho_{Hk}}{\rho_{Fl}} \\
1 - \frac{\cancel{\frac{\pi}{6}} \cdot h^2 \cdot (3 \cdot d_a - 2 \cdot h)}{\cancel{\frac{\pi}{6}} \cdot d_a^3} &= \frac{\rho_{Hk}}{\rho_{Fl}} \\
1 - 3 \cdot \left(\frac{h}{d_a}\right)^2 + 2 \cdot \left(\frac{h}{d_a}\right)^3 &= \frac{\rho_{Hk}}{\rho_{Fl}} \quad (**)
\end{aligned}\right\} \rightarrow h, d_a, \rho_{Hk} \text{ und } \rho_{Fl} \text{ darstellen}$$

Damit lässt sich f als Funktion in Abhängigkeit von

$$f(h, d_a, \rho_{Hk}, \rho_{Fl}) = 2 \cdot \left(\frac{h}{d_a}\right)^3 - 3 \cdot \left(\frac{h}{d_a}\right)^2 + 1 - \frac{\rho_{Hk}}{\rho_{Fl}}$$

Mit $\rho_{Hk} = \rho_W - \left(\frac{d_i}{d_a}\right)^3 \cdot \left(\rho_W - \frac{\rho_G}{1000}\right)$ erhält man folgende Funktion mit mehreren Veränderlichen:

$$f(h, d_a, d_i, \rho_w, \rho_G, \rho_{Fl}) = 2 \cdot \left(\frac{h}{d_a}\right)^3 - 3 \cdot \left(\frac{h}{d_a}\right)^2 + 1 - \frac{\rho_w - \left(\frac{d_i}{d_a}\right)^3 \cdot \left(\rho_w - \frac{\rho_G}{1000}\right)}{\rho_{Fl}}$$

$$d_i < d_a < d_{as}$$

wobei für d_a gilt:

$$d_i < d_a < \frac{d_i}{10 \cdot \sqrt[3]{\frac{\rho_W - 1}{1000 \cdot \rho_W - \rho_G}}}$$

Mit den auf den Seiten 22 und 24 angegebenen Werten erhält man

$$f(h, d_a) = 2 \cdot \left(\frac{h}{d_a}\right)^3 - 3 \cdot \left(\frac{h}{d_a}\right)^2 + \frac{7848.71 - 6.85 \cdot d_a^3}{d_a^3}$$

Die Nullstelle dieser Funktion für ein d_a in dem o. a. Bereich gibt die Höhe des aus dem Wasser ragenden Kugelabschnitts an; für $d_a = 10.4$ erhält man h = 2.324 dm und damit eine Eintauchtiefe von $t = d_a - h = 8.076\ dm$

See picture on page 23.

For calculating the immersion depth we equate weight and lifting force like above:
weight = lifting force or lifting force − weight = 0 and then solve for *h*.

$$vu(d,h) := vs(d) - \pi \cdot h^2 \cdot \left(\frac{d}{2} - \frac{h}{3}\right) \text{ Done}$$

$$\rho l \cdot vu(do,h) \cdot g - phs(do,di,\rho g,\rho m) \cdot vs(do) \cdot g = 0$$

$$\frac{g \cdot \left(2 \cdot h^3 \cdot \rho l - 3 \cdot h^2 \cdot do \cdot \rho l - di^3 \cdot (\rho g - \rho m) + do^3 \cdot (\rho l - \rho m)\right) \cdot \pi}{6} = 0$$

$$f(h,do,di,\rho m,\rho g,\rho l) := 2 \cdot h^3 \cdot \rho l - 3 \cdot h^2 \cdot do \cdot \rho l - di^3 \cdot (\rho g - \rho m) + do^3 \cdot (\rho l - \rho m) \text{ Done}$$

The zero of this function gives the height above the level of liquid.

We test the function using the data from above:

$$zeros\left(f\left(h \cdot \_dm, 10.4 \cdot \_dm, 10 \cdot \_dm, \frac{7.85 \cdot \_kg}{\_dm^3}, \frac{1.293 \cdot \_kg}{\_m^3}, \frac{1 \cdot \_kg}{\_dm^3}\right), h\right)$$

$$\{-2.01731, 2.32384, 15.2935\}$$

$$t = 10.4 - 2.32384 \quad t = 8.07616$$

Compare with the results above!

For *do* must hold:

$$d_i < do < do\_max$$

$$di < do < di \sqrt[3]{\frac{\rho m - \rho g}{\rho m - \rho l}}$$

We left the specific weight of the liquid as an argument of function *f* in order to be more flexible. So, we can not only vary the radii and the material of the sphere but also the liquid.

It works, what about swimming in the Dead Sea (ρl = 1.240 kg/dm³)?

$$zeros\left(f\left(h \cdot \_dm, 10.4 \cdot \_dm, 10 \cdot \_dm, \frac{7.85 \cdot \_kg}{\_dm^3}, \frac{1.293 \cdot \_kg}{\_m^3}, \frac{1.24 \cdot \_kg}{\_dm^3}\right), h\right)$$

$$\{-2.99399, 3.75047, 14.8435\}$$

$$10.4 - 3.75047 \quad 6.64953$$

3.75 *dm* are above water level and 6.65 *dm* are under water.

Now we throw the sphere into a sea of olive oil (ρl = 910 kg/m³).
(https://de.wikibooks.org/wiki/Tabellensammlung_Chemie/_Dichte_flüssiger_Stoffe)

$$zeros\left(f\left(h \cdot \_dm, 10.4 \cdot \_dm, 10 \cdot \_dm, \frac{7.85 \cdot \_kg}{\_dm^3}, \frac{1.293 \cdot \_kg}{\_m^3}, \frac{910 \cdot \_kg}{\_m^3}\right), h\right)$$

$$\{-1.175, 1.27135, 15.5036\}$$

1.27 *dm* are above and 9.13 *dm* are below of the surface.

Try with gasoline (ρl = 750 kg/m³).

Grafische Veranschaulichung mit GeoGebra

Betrachtet wird dazu (siehe S. 28) die Gleichung $2\cdot\left(\frac{h}{d_a}\right)^3-3\cdot\left(\frac{h}{d_a}\right)^2+1=\frac{\rho_{Hk}}{\rho_{Fl}}$ und damit ρ_{HK} als Funktion von *h* und ρ_{Fl}. Die Grafik unten stellt

$$\rho_{HK}(h,\rho_{Fl})=\rho_{Fl}\cdot\left(2\cdot\left(\frac{h}{d_a}\right)^3-3\cdot\left(\frac{h}{d_a}\right)^2+1\right)$$

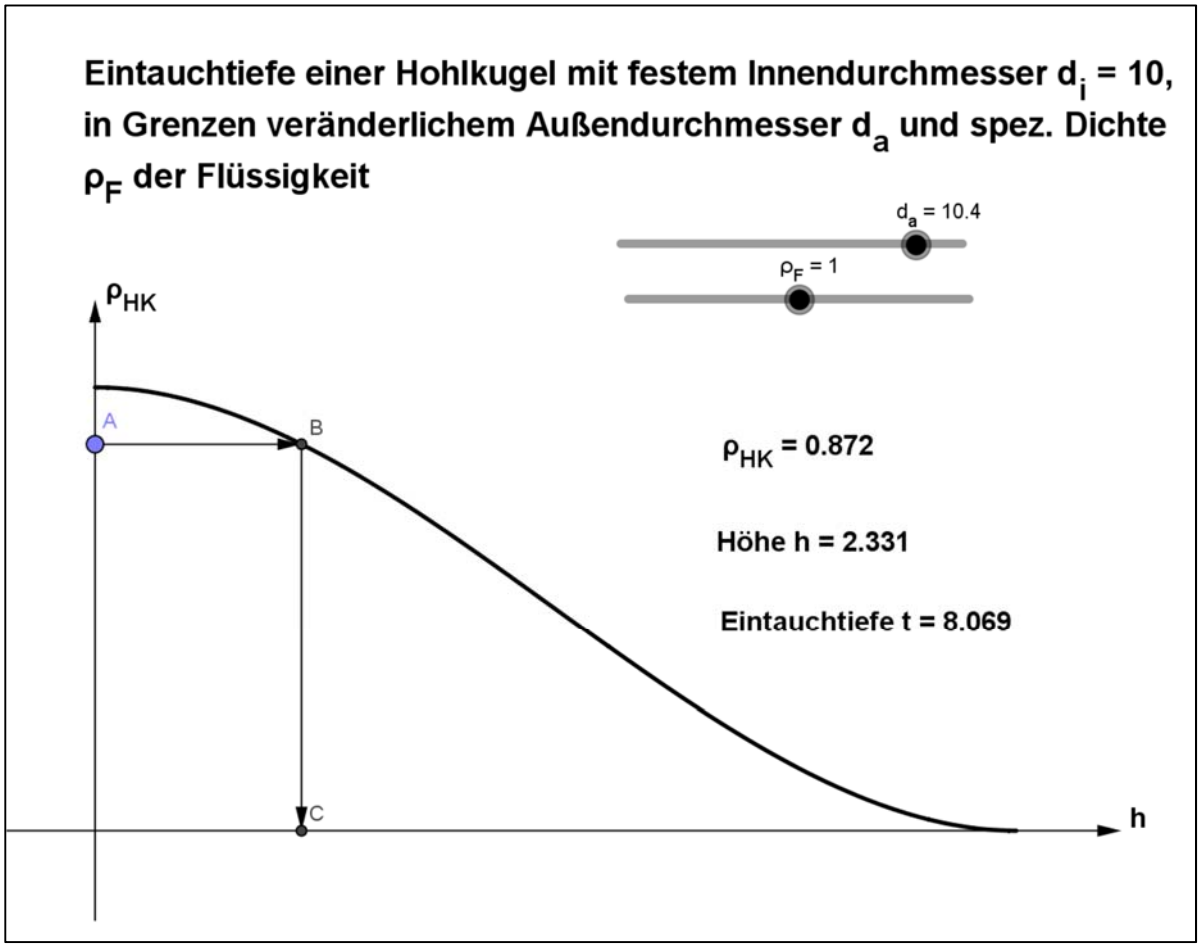
dar; ρ_{Fl} wird über einen Schieberegler eingestellt. Die Grafik zeigt die Abhängigkeit des spezifischen Gewichts der Hohlkugel von der Höhe des herausragenden Teils der Hohlkugel aus der Flüssigkeit.

Ausgehend von einem spez. Gewicht kann man aus der Grafik die Höhe *h* = *x*(*C*)

des herausragenden Teils der Hohlkugel ablesen. Die Eintauchtiefe beträgt $t = d_a - h$.

Der Außendurchmesser d_a und das spezifische Gewicht der Flüssigkeit lassen sich in Grenzen verändern.

Durch Ziehen des Punktes A verändert man das spez. Gewicht der Hohlkugel; die Ergebnisse werden angezeigt.



Diese Position ergibt die Werte von Seite 28.

Graphic representation with TI-Nspire

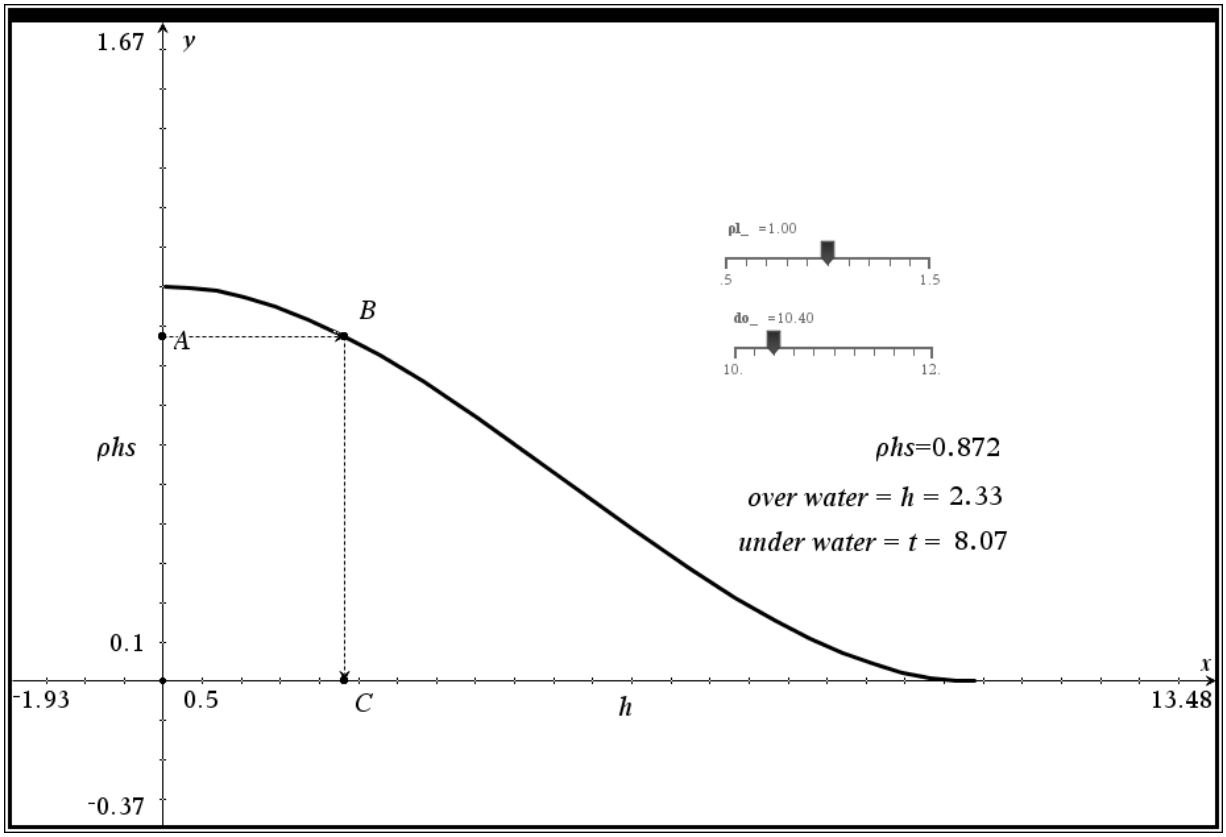
We define the specific weight of the hollow sphere ρhs as a function of h and the specific weight of the liquid ρl which will be controlled by a slider.

$$\text{solve}\left(\rho l\cdot \mathbf{vu}(do,h)=\rho hs\_\cdot \mathbf{vs}(do),\rho hs\_\right) \blacktriangleright \rho hs\_=\frac{\left(2\cdot h^3-3\cdot h^2\cdot do+do^3\right)\cdot \rho l}{do^3}$$

$$\mathbf{f1}(x):=\left\{\frac{\left(2\cdot h^3-3\cdot h^2\cdot do+do^3\right)\cdot \rho l}{do^3},0\le x\le \mathbf{do\_}\ \middle|\ h=x \text{ and } do=\mathbf{do\_} \text{ and } \rho l=\boldsymbol{\rho l}\_\right. \blacktriangleright \text{Done}$$

A second slider controls the outer diameter of the sphere.

By dragging point *A* we can vary the specific weight of the sphere. The *x*-coordinate of point *C* gives the height above the water line.



The results are displayed in a text box.

We show the position based on the values given on page 29.

Deriving the formula for $\sum_{k=1}^{N} k^p$ for any p, part 1

Don Speray, USA

Don Speray sent a mail:

> Hi Josef,
>
> Attached is the TNS file, along with the PDF file with small improvements. I mentioned that the code also works for the 92+, but I forgot to say the assignments must change from the TI-Nspire syntax to the older 92+ syntax.
>
> I use Mathematica for writing because I don't have a way of writing math in Word any longer. This caused my delay because Mathematica stopped working several months ago (a licensing bug), but all is fine now. I also have a new license for the TI-Nspire Student Software that runs on all my computers. So, although I own many TI-Nspire CAS (touch-pad and CX, and a box of early prototypes) calculators, I actually do this work on my PC. As I mentioned in my first letter to you, I collect calculating devices of all kinds, including calculators. Because I like the TI-92+ so much, I have been buying all I can to save them for posterity (and TI-89s and TI-898 Titaniums, too). I currently have more than 40 92+s – my home is a calculator museum! I use a 92+ regularly for programming probability simulations. I should use Mathematica since it cost so much, but the 92+ is so convenient for many tasks. I have HP calculators, too, and I am slowly learning all their symbolic math capabilities, from the past few decades up to the Prime, in order to see how computer algebra systems on handheld calculators have evolved.
>
> Thanks for your time and efforts,
> Don

For many years, I have enjoyed discovering new ways to derive the formula for the sum of the first N integers raised to a power p. I'm no Jacob Bernoulli, who solved this over 300 years ago, but I've got a CAS! In this series, I will present several simple ways that use today's calculators to derive the formula for the sum for an arbitrary p. (Note: these methods are original to me, but I do not claim originality!) This first method, which I call the recursive way, bootstraps from known expressions for the sums of previous values of p. That is, once $\sum_{k=1}^{N} k^q$, q = 1, 2, 3, …, p-1 are known, they are used to derive $\sum_{k=1}^{N} k^p$.

We begin with the seemingly wrong step of writing the sum for $p + 1$, and in a roundabout way. We'll see that this is just a ploy to introduce into the final expression.

$$\sum_{k=1}^{N} k^{p+1} = \sum_{k=0}^{N} (k+1)^{p+1}$$

$$= \sum_{k=0}^{N-1} k^{p+1} + \binom{p+1}{p} \sum_{k=0}^{N-1} k^p + \binom{p+1}{p-1} \sum_{k=0}^{N-1} k^{p-1} + \ldots + \binom{p+1}{1} \sum_{k=0}^{N-1} k^0 + \sum_{k=0}^{N-1} 1$$

Note that each sum in the binomial expansion is just short of the full sum to N, so let's add the final term to each one (to both sides of the equation, of course). The last sum is simply N, so we'll leave it alone. We can also change the lower limit to 1. Note that the first sum in the expansion then becomes equal to the original left-hand side, and they cancel, leaving:

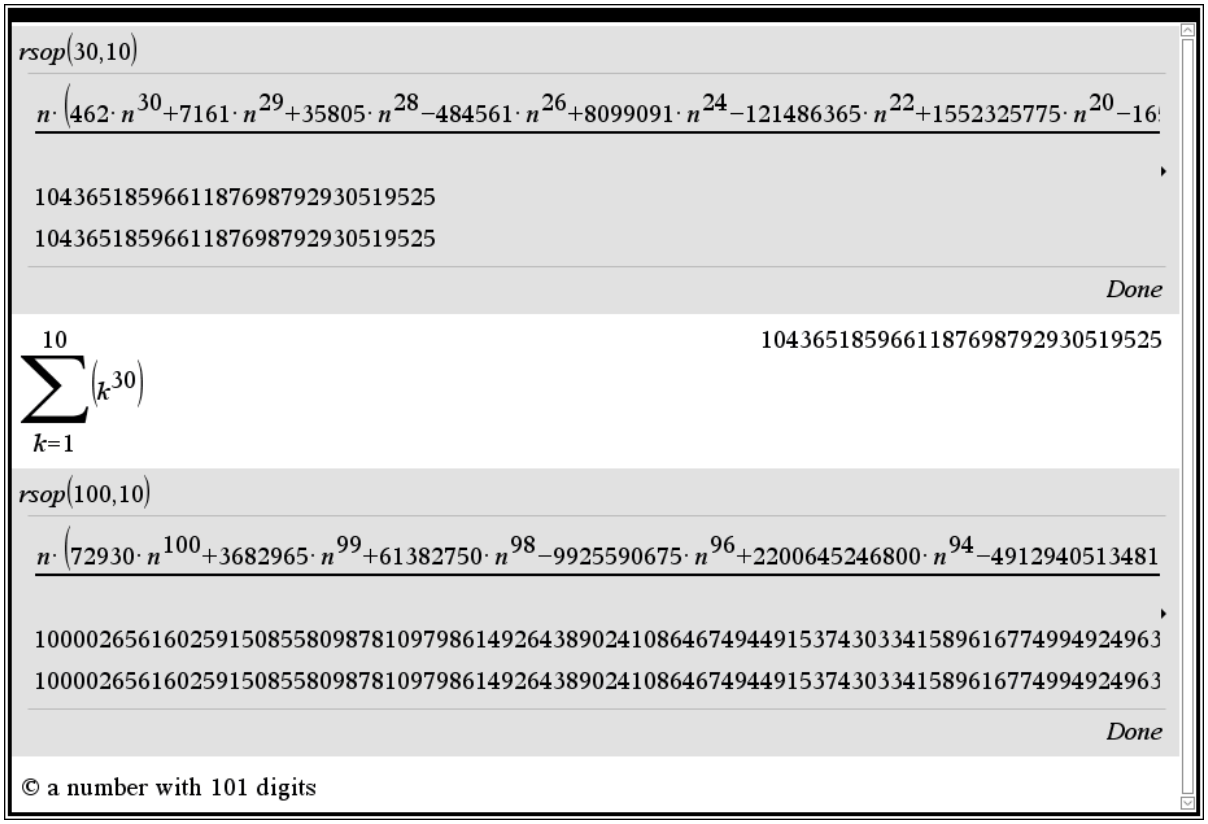
$$N^{p+1}+\binom{p+1}{p}N^{p}+\binom{p+1}{p-1}N^{p-1}+\ldots+\binom{p+1}{1}$$
$$=\binom{p+1}{p}\sum_{k=1}^{N}k^{p}+\binom{p+1}{p-1}\sum_{k=1}^{N}k^{p-1}+\ldots+\binom{p+1}{1}\sum_{k=1}^{N}k+N$$

The first sum on the right-hand side is what we're looking for. The other sums will have been previously derived during this bootstrapping process. Solving for it:

$$\sum_{k=1}^{N}k^{p}=\frac{N^{p+1}+\binom{p+1}{p}N^{p}-N+\sum_{j=1}^{p-1}\left(\binom{p+1}{j}\left(N^{j}-\sum_{k=1}^{N}k^{j}\right)\right)}{\binom{p+1}{p}}$$

Note that the formula is a polynomial of degree $p + 1$. We'll make use of this in later methods. Here is a simple, straight-forward TI-Nspire (and 92+) program for symbolically computing the formula. Although I call it recursive, the code uses iteration to construct the sum formulas, starting with $p = 1$, and appends them to a list, ready for constructing the next. Use it as **`rsop(p,upper_limit)`**, where $p \geq 1$, and `upper_limit` is used to check the result. Below the code is the result of **`rsop(30,10)`**. Be sure to check out the result for $p = 100$.

```
"rsop" stored successfully
Define LibPub rsop(final_p,upper_lim)=
Prgm
Local p,j,s,sums
DelVar n   © force symbolic computation
sums:={}
For p,1,final_p
  s:=n^(p+1)+nCr(p+1,p)·n^p−n
  For j,1,p−1
    s:=s+nCr(p+1,j)·(n^j−sums[j])
  EndFor
  s:=s/nCr(p+1,p):  sums:=augment(sums,{s})
EndFor
© Final 's' is what we're after.
Disp s
©  Numerically verify:
Disp s|n=upper_lim
Disp Σ(k=1, upper_lim) (k^final_p)
EndPrgm
```



This is the sum in full length:

((n*(462*n^(30)+7161*n^(29)+35805*n^(28)−484561*n^(26)+8099091*n^(24)−
121486365*n^(22)+1552325775*n^(20)−16502417085*n^(18)+142933380975*n^(16)−
984742931403*n^(14)+5238144213225*n^(12)−20698604632251*n^(10)+57673154564025*n^(8)−
105183202315455*n^(6)+111901503855141*n^(4)−
56689963476223*n^(2)+8615841276005))/(14322))

A Bug or Something?

In the second part of this series I will mention a bug (or something) I have found with function linSolve. When you call it with an explicit list of unknowns, like:

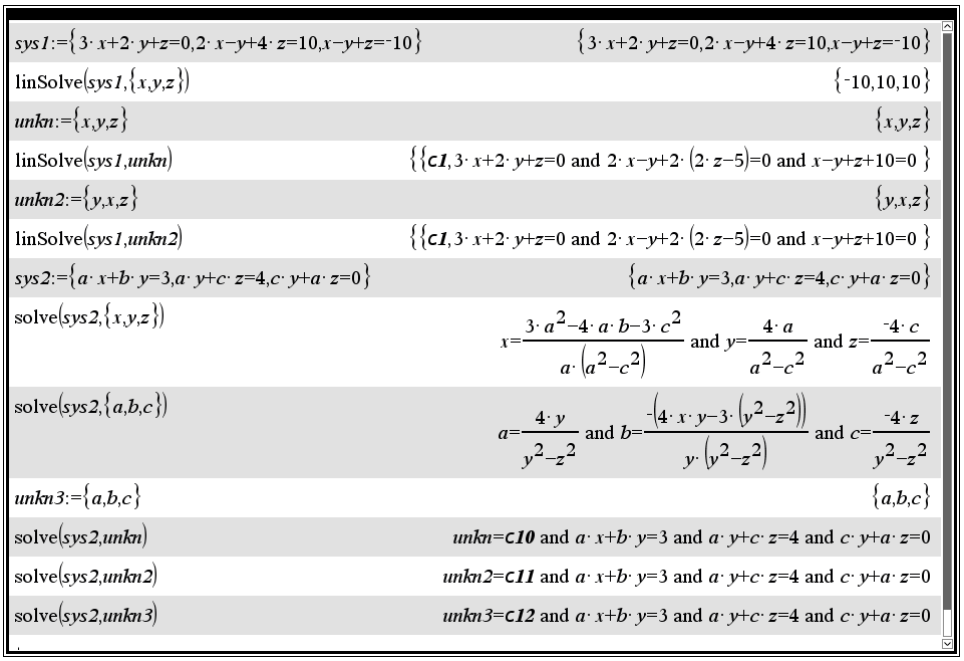
linSolve (equations, {x, y, z})

it returns a list of solutions. But if you use a list variable for the unknowns, like

unknowns := {x, y, z}
linSolve (equations, unknowns)

it does something completely different. In my program, I have a work-around, but it is ugly.
Any advice?

The next page shows the TI-Nspire screen compared with the DERIVE output, Josef.



$sys1:=\{3\cdot x+2\cdot y+z=0,2\cdot x-y+4\cdot z=10,x-y+z=^{-}10\}$	$\{3\cdot x+2\cdot y+z=0,2\cdot x-y+4\cdot z=10,x-y+z=^{-}10\}$
$\text{linSolve}(sys1,\{x,y,z\})$	$\{^{-}10,10,10\}$
$unkn:=\{x,y,z\}$	$\{x,y,z\}$
$\text{linSolve}(sys1,unkn)$	$\{\{c1,3\cdot x+2\cdot y+z=0 \text{ and } 2\cdot x-y+2\cdot(2\cdot z-5)=0 \text{ and } x-y+z+10=0\}$
$unkn2:=\{y,x,z\}$	$\{y,x,z\}$
$\text{linSolve}(sys1,unkn2)$	$\{\{c1,3\cdot x+2\cdot y+z=0 \text{ and } 2\cdot x-y+2\cdot(2\cdot z-5)=0 \text{ and } x-y+z+10=0\}$
$sys2:=\{a\cdot x+b\cdot y=3,a\cdot y+c\cdot z=4,c\cdot y+a\cdot z=0\}$	$\{a\cdot x+b\cdot y=3,a\cdot y+c\cdot z=4,c\cdot y+a\cdot z=0\}$
$\text{solve}(sys2,\{x,y,z\})$	$x=\frac{3\cdot a^2-4\cdot a\cdot b-3\cdot c^2}{a\cdot(a^2-c^2)} \text{ and } y=\frac{4\cdot a}{a^2-c^2} \text{ and } z=\frac{^{-}4\cdot c}{a^2-c^2}$
$\text{solve}(sys2,\{a,b,c\})$	$a=\frac{4\cdot y}{y^2-z^2} \text{ and } b=\frac{^{-}(4\cdot x\cdot y-3\cdot(y^2-z^2))}{y\cdot(y^2-z^2)} \text{ and } c=\frac{^{-}4\cdot z}{y^2-z^2}$
$unkn3:=\{a,b,c\}$	$\{a,b,c\}$
$\text{solve}(sys2,unkn)$	$unkn=c10 \text{ and } a\cdot x+b\cdot y=3 \text{ and } a\cdot y+c\cdot z=4 \text{ and } c\cdot y+a\cdot z=0$
$\text{solve}(sys2,unkn2)$	$unkn2=c11 \text{ and } a\cdot x+b\cdot y=3 \text{ and } a\cdot y+c\cdot z=4 \text{ and } c\cdot y+a\cdot z=0$
$\text{solve}(sys2,unkn3)$	$unkn3=c12 \text{ and } a\cdot x+b\cdot y=3 \text{ and } a\cdot y+c\cdot z=4 \text{ and } c\cdot y+a\cdot z=0$

Notice the strange difference in the "results" of lines 4 and 6 and the last three lines.

#1: `sys := [a·x + b·y = 3, a·y + c·z = 4, c·y + a·z = 0]`

#2: `[vars1 := [x, y, z], vars2 := [z, x, y], vars3 := [a, b, c], vars4 := [a, x, y]]`

#3: $$\text{SOLVE(sys, vars1)} = \left[x = \frac{3\cdot a^2 - 4\cdot a\cdot b - 3\cdot c^2}{a\cdot(a^2 - c^2)} \wedge y = \frac{4\cdot a}{a^2 - c^2} \wedge z = \frac{4\cdot c}{c^2 - a^2}\right]$$

#4: $$\text{SOLVE(sys, vars2)} = \left[z = \frac{4\cdot c}{c^2 - a^2} \wedge x = \frac{3\cdot a^2 - 4\cdot a\cdot b - 3\cdot c^2}{a\cdot(a^2 - c^2)} \wedge y = \frac{4\cdot a}{a^2 - c^2}\right]$$

#5: $$\text{SOLVE(sys, vars3)} = \left[a = \frac{4\cdot y}{y^2 - z^2} \wedge b = \frac{4\cdot x\cdot y - 3\cdot(y^2 - z^2)}{y\cdot(z^2 - y^2)} \wedge c = \frac{4\cdot z}{z^2 - y^2}\right]$$

#6: $$\text{SOLVE(sys, vars4)} = \left[a = \frac{c\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}}{z} \wedge x = \frac{\left(b\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}} + 3\right)\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}}{c\cdot z - 4} \wedge y = -\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}, a = -\frac{c\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}}{z} \wedge x = \frac{\left(b\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}} - 3\right)\cdot\sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}}{c\cdot z - 4} \wedge y = \sqrt{\frac{z\cdot(c\cdot z - 4)}{c}}\right]$$

I sent this request to several TI-Nspire users. Some of them considered this as a bug. One colleague from Germany does not share this opinion, he wrote:

I don't consider this behavior as a bug. It seems to be a consequence of the Nspire-syntax:
list {x,y,z} is treated as a variable which does not appear in the equations. linSolve and solve as well allow lists of more variables as arguments, which are given in a row, but they don't correspond with the data type "list". This makes the difference.

At the other hand, the data types of linSolve and solve differ significantly (list vs condition(s)). This is not an error, but can irritate sometimes.

The following problem was sent by Guiseppe Ornaghi, Italy:

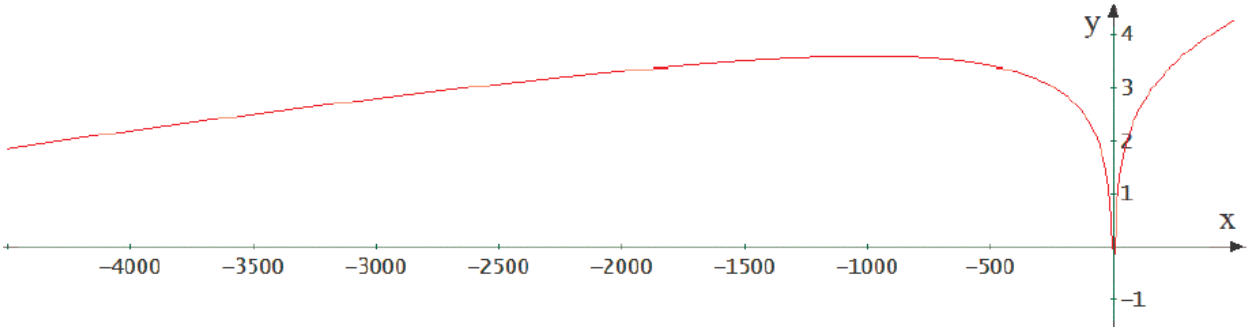
Dear Josef,
in DERIVE I noticed a strange behavior in the numerical solution of an equation, as explained in the attachment.

An amazing algorithm

Everything has begun with the function

$$y = f(x) = 0.43\left[\ln\left(x^2 - \frac{x}{2}\right) + 0.002x - 1.5\right]$$

enough represented by the following picture.



The coordinates of the maximum are (– 999.7500625, 3.580884513), thus a straight-line y = k intersects the curve in three different points, two of negative abscissa and one of positive abscissa, when k < 3.580884513, while in only one point, of positive abscissa, when k > 3.580884513. The abscissas of these intersections satisfy the equation

$$0.43\left[\ln\left(x^2 - \frac{x}{2}\right) + 0.002x\right] - 1.5 = k$$

of which the command "Solve Expression", with the options: *numeric method – domain of solution Real*, provides only one solution, whichever is the value of k, not surprising, as it is a transcendental equation.

The other possible solutions can be obtained by replacing, in the second option, *Real* with *Interval*, since the location of an appropriate interval can be easily identified on the graphical representation.

The surprising fact is that, when the solutions are three, the algorithm chooses the one to be supplied according to the criterion, as unexpected as it is unexplained:

Fixed the number σ = 2.728391135 ..., if k <σ the greater of the two negative solutions is chosen, while if k> σ, the positive solution is chosen.

#1: $f(x) := 0.43 \cdot \left(LN\left(x^2 - \frac{x}{2} \right) + 0.002 \cdot x \right) - 1.5$

#2: NSOLVE(f(x) = 2, x) = (x = -62.04097747)

#3: NSOLVE(f(x) = 2.72, x) = (x = -158.1508493)

#4: NSOLVE(f(x) = 2.73, x) = (x = 121.4191632)

#5: NSOLVE(f(x) = 3, x) = (x = 159.8553423)

Following this procedure, the questions are unavoidable:

1) Which motive suggests it and what is the meaning of σ?

2) Why not choose the solution belonging to the intersected branch, whichever k?

3) Why σ and not another number, for example *e*, close to it but far well known?

The following table shows the determination of σ by means of contiguous classes, as well as the abscissas of the three intersections in the first cases.

(e.g. when y=k=2 then the three solutions are -4295.6..; -62.04..; 55.62.., when k=3 they are -2649.7.; -237.13..; 159.85.. etc)

y_-	y_+	$x(y_-)$			$x(y_+)$		
2	3	– 4295.648574	– 62.04097747	55.62627792	– 2649.760818	– 237.1381759	159.8553423
2.7	2.8	– 3181.449985	– 153.8446235	117.7011013	– 3009.667968	– 176.8794250	130.5052864
2.72	2.73	– 3147.450828	– 158.1508493	120.1682940	– 3130.387187	– 160.3574737	121.4191632
2.728	2.729	– 3133.803382	– 159.9132364	121.1680623	– 3132.095500	– 160.1351719	121.2935548
2.7283	2.7284	– 3133.291064	– 159.9797789	121.2056978	– 3133.120282	– 160.0019669	121.2182453

y_-	y_+	$x(y_-)$	$x(y_+)$
2.728391	2.728392	– 159.9999698	121.2172415
2.7283911	2.7283912	– 159.9999920	121.2171411
2.72839113		– 159.9999986	
2.72839114	2.72839115	– 160	121.2171348
2,72839113	2.72839114	– 159.9999987006524	121.2171336153804
2.278391135	2.278391136	– 159.9999998101294	121.2171361249799

The two rows with a light blue background indicate the result obtained with the initial position «*Options Mode* ...» and the option: *Digits* 10. The option has therefore been changed in *Digits* 16.

With 16 digits, f (- 160) = 2.728391135855675 ... It is very probable therefore that it is

$$\sigma = f(-160)$$

but why?

I'd like to compare how various CAS are treating this problem. Obviously, we will start with DERIVE. Supported by the function graph (page 36) – this is necessary in all systems – we can find the solutions without problems:

$$\#1: \quad f(x) := 0.43 \cdot \left(\mathrm{LN}\left(x^2 - \frac{x}{2} \right) + 0.002 \cdot x \right) - 1.5$$

#2: NSOLVE(f(x) = 3, x) = (x = 159.8553423)

#3: NSOLVE(f(x) = 3, x, −500, −100) = (x = −237.1381759)

#4: NSOLVE(f(x) = 3, x, −5000, −1000) = (x = −2649.761199)

#5: NSOLVE(f(x) = 2, x) = (x = −62.04097747)

#6: NSOLVE(f(x) = 2, x, 10, 100) = (x = 55.62628507)

#7: NSOLVE(f(x) = 2, x, −5000, −100) = (x = −4295.648598)

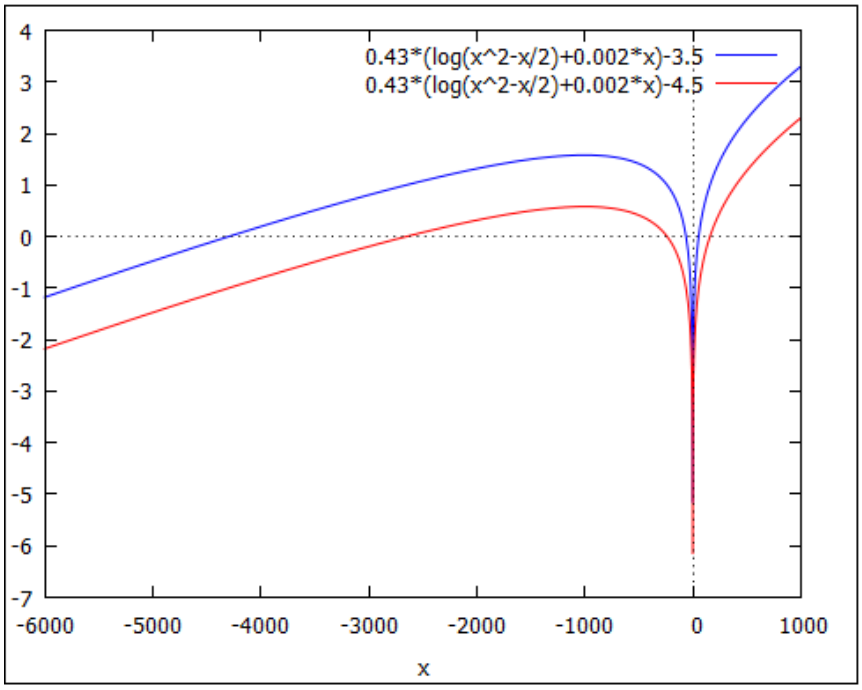
How does Maxima behave?

Leon Magiera from Poland wrote: "I believe that this method is a "little" primitive!".

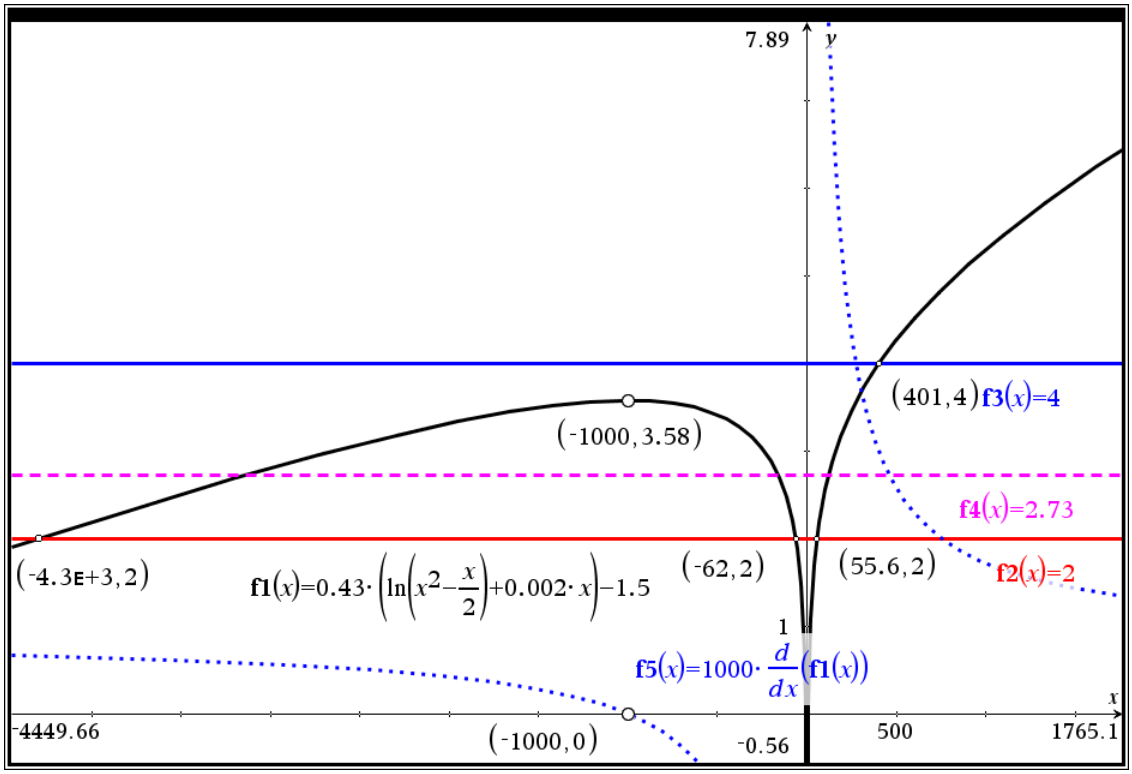
Regards, Leon

```
(%i1)    g(k):=0.43*(log(x^2-x/2)+0.002*x)-1.5-k$
(%i3)    plot2d([g(2),g(3)],[x,-6000,1000]);
-->      find_root(g(2),x,50,60);
(%o7)    55.62628049350963
-->      find_root(g(2),x,-100,-1);
(%o8)    -62.04096961729991
-->      find_root(g(2),x,-5000,-4000);
(%o9)    -4295.648585668234
```

Figure 1:



Now let's turn to TI-NspireCAS:

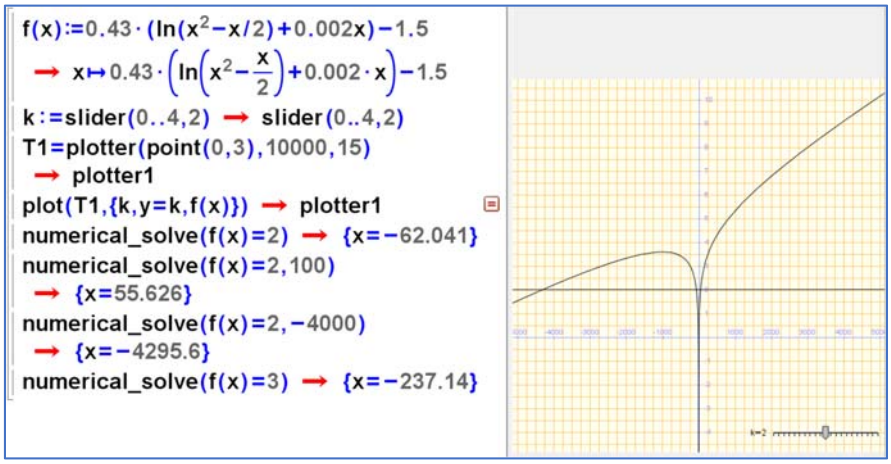


$f1(x):=0.43\cdot\left(\ln\left(x^2-\frac{x}{2}\right)+0.002\cdot x\right)-1.5$	*Done*
$\text{nSolve}(f1(x)=2,x)$	55.626280
$\text{nSolve}(f1(x)=2,x)\mid x<55.6$	-62.040970
$\text{nSolve}(f1(x)=2,x)\mid x<-62.1$	-4295.648586
$\text{nSolve}(f1(x)=4,x)$	401.289387
$f1(-160.)$	2.728391
$f5(x):=\frac{d}{dx}(f1(x))$	*Done*
⚠ $\text{nSolve}(f5(x)=0,x)$	0.250062
⚠ $\text{nSolve}(f5(x)=0,x)\mid x<0$	-999.750062
$\text{nSolve}(f1(x)=3,x)$	159.855324

We see that TI-Nspire gives the positive solution and offers one solution without entering intervals or approximations derived from the function graph.

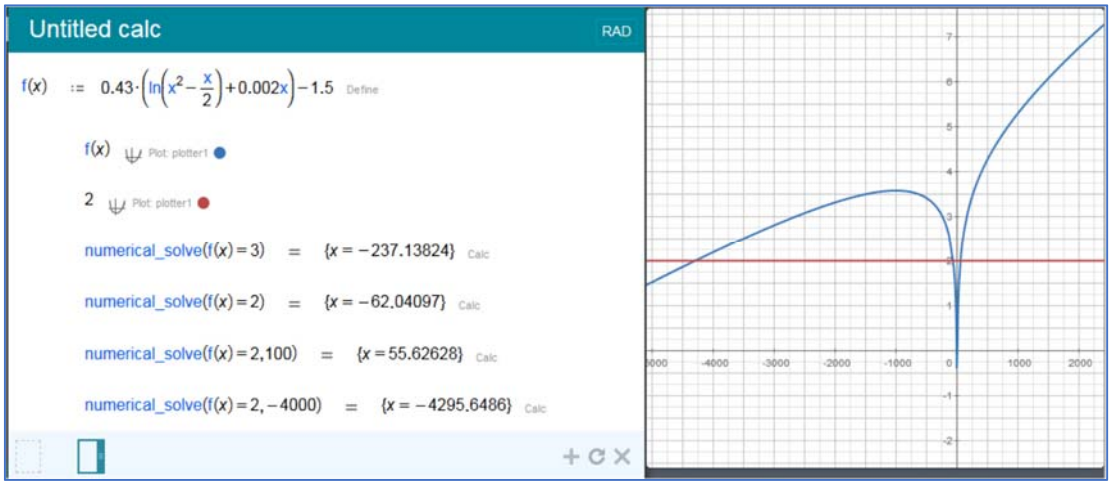
Then I tried WIRIS:

I couldn't install my old WIRIS on my computer but thanks great help of the WIRIS-support I got the latest version of WIRIS together with the respective license key.



Message from WIRIS-support:

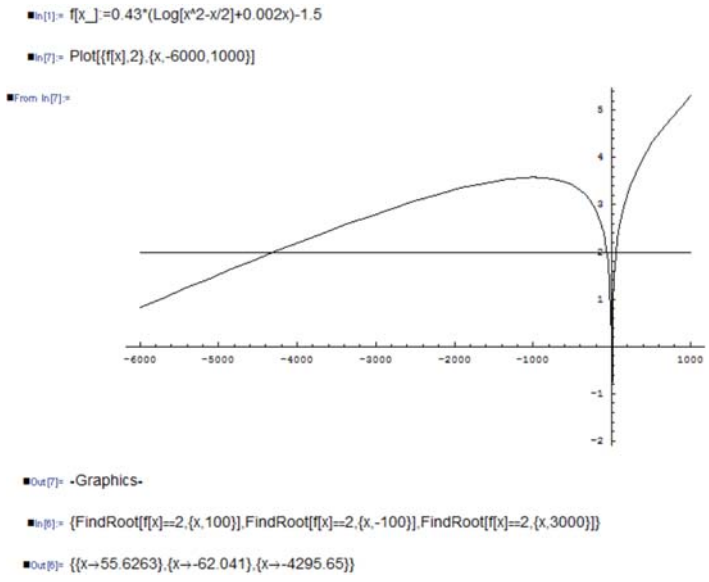
I'm sorry, but we don't support Wiris CAS Desktop anymore and I was told that we can't make any new license keys for it anymore either. Our current CAS product is https://calcme.com , which you can use online for free. (Old WIRIS file can be opened).



I recommend trying CalcMe, Josef.

My last try was MATHEMATICA (an old version):

It works similar to Maxima, needing approximations found in the graph.



One of Hubert Langlotz's, Germany, students tried to solve a problem (arc length of a curve) performing the calculation as depicted.
Hubert wrote: I was surprised that it is possible to have roots as borders of a sum.

$$\sum_{a=0}^{\sqrt{2}}\left(1-\frac{a^2}{2}\right)-\sum_{a=\sqrt{2}}^{2.5}\left(1-\frac{a^2}{2}\right) \qquad 3.23917$$

Helmut Heugl and I tried and we found out that we could receive four different results depending on the calculation mode (see below).

$\sum_{x=0}^{\sqrt{2}}(x)$	$\frac{\sqrt{2}}{2}+1$
$\sum_{x=0}^{\sqrt{2.}}(x)$	1.00000000000
$\sum_{x=0}^{\sqrt{2}}(x)-\sum_{x=\sqrt{2}}^{\frac{5}{2}}(x)$	$\frac{-19}{8}$
$\sum_{x=0}^{\sqrt{2}}(x)-\sum_{x=\sqrt{2}}^{\frac{5.}{2}}(x)$	-2.12132034356
$f(a):=1-\frac{a^2}{2}$	*Done*

DERIVE gives only one result (explained by its great feature "Stepwise Simplification":

```
#1:   Σ (x=√2 to 4) x

      Σ (x=a to b) F(x) → F(a) + F(a + 1) + ... + F(b)

#2:   √2 + √2 + √2 + 3
```

$\sum_{a=0}^{\sqrt{2.}}(f(a))-\sum_{a=\sqrt{2}}^{\frac{5}{2}}(f(a))$	3.699957911 38
$\sum_{a=0}^{\sqrt{2.}}(f(a))-\sum_{a=\sqrt{2}}^{2.5}(f(a))$	3.41421356237
$\sum_{a=0}^{\sqrt{2}}(f(a))-\sum_{a=\sqrt{2}}^{2.5}(f(a))$	3.23917147376
$\sum_{a=0}^{\sqrt{2}}(f(a))-\sum_{a=\sqrt{2}}^{\frac{5}{2}}(f(a))$	$\frac{7\cdot\sqrt{2}}{6}+\frac{15}{8}$
$\frac{7\cdot\sqrt{2}}{6}+\frac{15.}{8}$	3.52491582277

```
#1:   Σ (x=0 to √2) x

#2:   1

#3:   1

#4:   (Σ (x=0 to √2) x) - Σ (x=√2 to 5/2) x

#5:   - 2·√2

#6:   -2.828427124

#7:   f(a) := 1 - a^2/2

#8:   (Σ (a=0 to √2) f(a)) - Σ (a=√2 to 5/2) f(a)

#9:   √2 + 2

#10:  3.414213562
```

Mail from Miroslav Kures:

Dear Josef,

I send you link to my new puzzle page. I hope DUG-members can enjoy it.

https://searchfornumbers.blogspot.com/

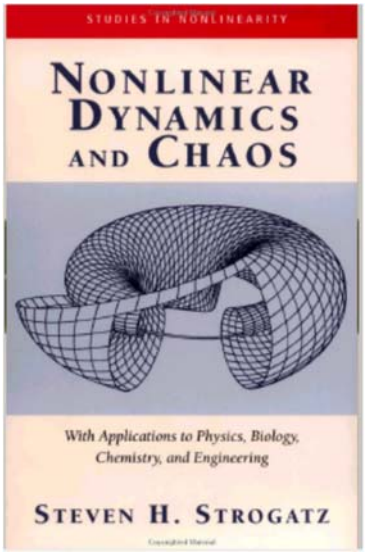
All the best,

Miroslav Kures

Our member David Dyer mentioned in one of his mails *Nonlinear Dynamics and Chaos*, written by *Steven H. Strogatz*. I found out that you can download this book (257 pages):

http://detritus.fundacioace.com/pub/books/Strogatz%20-%20Nonlinear%20dynamics%20and%20chaos.pdf

There are some other complete books on Mathematical Biology which you can download for free:



J. D. Murray, *Mathematical Biology*
http://www.ift.unesp.br/users/mmemezes/mathbio.pdf

Brian Ingalls, *Mathematical Modelling in System Biology*
https://www.math.uwaterloo.ca/~bingalls/MMSB/Notes.pdf

Christina Kuttler, *Mathematical Models in Biology*
http://www-m6.ma.tum.de/~kuttler/script1.pdf

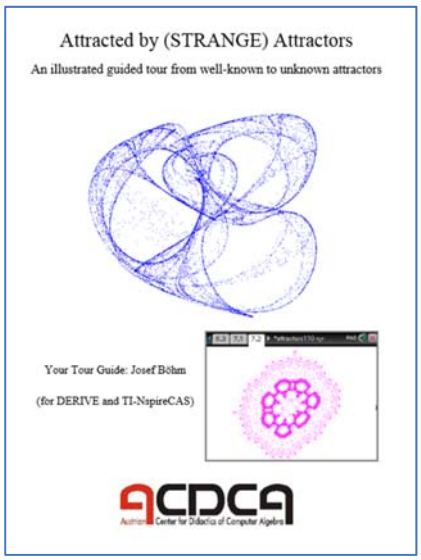
Jeffrey R. Chasnov, *Mathematical Biology*
https://www.math.ust.hk/~machas/mathematical-biology.pdf

I was so fascinated by the "Strange Attractors" that I couldn't resist to put all my respective contributions together to one paper accomplished with some add-ons.

You can download the paper (83 pages) together with all DERIVE- and TI-Nspire files from

http://rfdz.ph-noe.ac.at/acdca/materialien.html

Josef



This is the first of Don' Goodies for TI-Nspire. Two more will follow with the next DNL issues.

GSolve is a function I wrote to handle solving multiple (or singular) equations easily. All you have to do is enter the equation(s) with the unknown(s) to be solved for. For instance:

gsolve$(l\cdot 5=40)$ ▸ $l=8$

Or: **gsolve**$\left(\begin{cases} x+2\cdot y=-7 \\ 2\cdot x-3\cdot y=0 \end{cases}\right)$ ▸ $x=-3$ and $y=-2$

gsolve() will solve equations and inequalities with real answers. cgsolve() will solve for complex answers.

cgsolve$(p^2-2\cdot p+3=0)$ ▸ $p=1-\sqrt{2}\cdot i$ or $p=1+\sqrt{2}\cdot i$

cgsolve$\left(\begin{cases} p^2-2\cdot q+3=0 \\ q^2-2\cdot p=10 \end{cases}\right)$

▸ $p=-0.3116+3.0848\cdot i$ and $q=-3.2095-0.9612\cdot i$ or
$p=-0.3116-3.0848\cdot i$ and $q=-3.2095+0.9612\cdot i$ or $p=-1.5110$ and $q=2.6416$ or
$p=2.1342$ and $q=3.7773$

There is another way to use the gsolve function if you have a equation or set of equations you will use a lot. For instance, see the function for linearmotion on the next page. This uses **gsolve**() as a subroutine.

```
Define LibPub gsolve(eqs)=
Func
Local vars,dm,eqq,i_
vars:=varlist(eqs)
dm:=dim(vars)
eqq:="solve("&string(eqs)&",{"
For i_,1,dm
  If i_=1 Then
    eqq:=eqq&string(vars[i_])
  Else
    eqq:=eqq&","&string(vars[i_])
  EndIf
EndFor
eqq:=eqq&"})"
expr(eqq)
EndFunc
```

```
Define LibPub cgsolve(eqs)=
Func
Local vars,dm,eqq,i_
vars:=varlist(eqs)
dm:=dim(vars)
eqq:="csolve("&string(eqs)&",{"
For i_,1,dm
  If i_=1 Then
    eqq:=eqq&string(vars[i_])
  Else
    eqq:=eqq&","&string(vars[i_])
  EndIf
EndFor
eqq:=eqq&"})"
expr(eqq)
EndFunc
```

The functions I have written use the function varlist() which was written originally for the TI-89 and I adopted for the TI-NSpire CAS. It makes use of an undocumented function on the calculator--part(). part() is used to systematically breakup expressions into parts so variables can be isolated and put into a list.

```
Define LibPub varlist(xpr)=
Func
©XPR
Local tmp
If getType(xpr)="MAT"
  xpr:=mat▶list(xpr)
If getType(xpr)="LIST" Then
  If dim(xpr)=1
  xpr:=xpr[1]
EndIf
If when(part(xpr)=0,true,false,false)
  Return when(iscmplxn(xpr),{},{xpr},{undef})
If part(xpr)=1
  Return removedup(varlist(part(xpr,1)))
If when(part(xpr)≥2,true,false,false) Then
  tmp:=augment(varlist(part(xpr,1)),varlist(seq(part(xpr,ii),ii,2,part(xpr))))
  Return removedup(tmp)
EndIf
EndFunc
```

The functions I have written use the function varlist() which was written originally for the TI−89 by someone else and I adopted for the nSpire CAS. It makes use of an undocumented function on the calculator−−part(). part() is used to systematically breakup expressions into parts so variables can be isolated and put into a list.

dim() and part() work differently. dim() works on a list, matrix, or a string. part() works directly with an expression or an equation. When using simply part() it will determine the highest level operation in an expression or equation and divide it into 2 at that operation. For example,

$\text{part}(2\cdot x+3\cdot y=z) \blacktriangleright 2$ divides the equation into 2 parts at the = sign. Thus:

$\text{part}(2\cdot x+3\cdot y=z,1) \blacktriangleright 2\cdot x+3\cdot y$ and $\text{part}(2\cdot x+3\cdot y=z,2) \blacktriangleright z$

Now, you can take the first part and break it up more.

$\text{part}(2\cdot x+3\cdot y) \blacktriangleright 2$ has 2 parts with the plus sign as the highest level operation.

Continuing:

$\text{part}(\text{part}(2\cdot x+3\cdot y,1),2) \blacktriangleright x$

The function varlist() does this all step by step until it has a list of the variables, without duplicates, in the expression or equation.

dim() cannot do this. $\dim(2\cdot x+3\cdot y=z)$ ▸ Error: Invalid data type

$\dim(\text{"2x+3y=z"})$ ▸ 7

You would have to use the mid() function to take the string apart, use expr() to change it from a string, and test that it is a variable.

Regards,

Don

$$\dim\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\} \blacktriangleright 4 \quad \text{and} \quad \text{part}\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\} \blacktriangleright 4$$

but compare:

$$\dim\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\},1$$ does not work!!

$$\text{part}\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\},1 \blacktriangleright v=a\cdot t+v0 \quad \text{part}\left(\text{part}\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\},2\right),2 \blacktriangleright \frac{t\cdot(v+v0)}{2}+x0$$

$$\text{part}\left(\text{part}\left(\text{part}\left\{\begin{matrix} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0)\end{matrix}\right\},2\right),2\right),1 \blacktriangleright \frac{t\cdot(v+v0)}{2}$$ and so on …

```
Define LibPub iscmplxn(x)=
Func
©X
Local r,i
r:=real(1.·x)
i:=imag(x)
when(getType(x)="NUM" or getType(x)="EXPR" and getType(r)="NUM" and
  getType(i)="NUM",true,false)
EndFunc
```

```
Define LibPub removedup(l)=
Func
©LIST
Local i,n
For i,dim(l),2,-1
  For n,1,i−1
    If when(l[n]=l[i],true,false,false) Then
  l:=augment(left(l,i−1),mid(l,i+1))
  Exit
  EndIf
EndFor
EndFor
EndFunc
```

Define LibPub **linearmotion**(x0,x,v0,v,a,t)=Func ▸ *Done*

Local eq

$$eq:=\begin{cases} v=v0+a\cdot t \\ x=x0+\frac{1}{2}\cdot(v+v0)\cdot t \\ x=x0+v0\cdot t+\frac{1}{2}\cdot a\cdot t^2 \\ v^2=v0^2+2\cdot a\cdot(x-x0) \end{cases}$$

gsolve(eq)

EndFunc

In linearmotion(x0,x,v0,v,a,t), where x0 is initial distance, x is final distance, v0 is initial velocity, v is final velocity, a is acceleration and t is time, enter the known values in the appropriate variable and the variable names for unknown variables. See an example.

Problem: A racing car reaches a speed of 50 m/s. At this instant it decelerates uniformly and comes to rest 5 s later. (a) Determine the deceleration of the car. (b) How far does the car travel after starting to brake?

So: x0=0, x=?, v0=50, v=0, a=?, t=5

linearmotion(0,x,50,0,a,5) ▸ $x=125$ and $a=-10$