THE DERIVE - NEWSLETTER #18

ISSN 1990-7079

THE BULLETIN OF THE

DERIVE®

USER GROUP

Contents:

1 Letter of the Editor

2 Editorial - Preview

3 DERIVE User Forum

5 Electronic Mailbox

Cerdán, de Córdoba, Ginestar and Ratis
14 Evaluation of the Bessel Functions

Böhm & Reichel & Klingen
27 Cubic Splines

Otto Reichel
40 "Reverse" Discussion of a Curve

Johann Wiesenbauer
44 Titbits from Algebra and Number Theory (5)

revised Version 2008 June 1995

[1] Mathematik unterrichten mit DERIVE, Bernhard Kutzler
Addison – Wesley, 1995, ISBN 3 89319 860 1

[2] Mathematisches Praktikum mit DERIVE, A. Garcia u.a., übersetzt von L. Klingen
Addison – Wesley, 1995, ISBN 3 89319 857 1

[3] Nuevas Tecnologías y Enseñanza de Matemáticas, A. Garcia, A. Martinez, R. Miñano
Editorial Sintesis, Madrid, 1995, ISBN 84 7738 283 2

[4] Nuevas Tecnologiás en Geometria, E. Roanes Macias & E. Roanes Lozano
Editorial Complutense, Madrid, 1994, ISBN 84 7491 531 7
(This is not a DERIVE book but it is a very interesting book written by father and son Roanes for everybody liking geometry. Muchas gracias for your book, Eugenio.)

[5] Handleiding DERIVE, P.E.J.M. Gondrie & G.A.T.M. van Alst
Academic Service, Schoonhoven, NL, 1994, ISBN 90 395 0080 0

[6] Toegepaste computeralgebra met DERIVE, C. de Ritter & A.K.H. Overeem-Loohuis
Academic Service, Schoonhoven, NL, 1994, ISBN 90 395 0194 7

[7] Toegepaste wiskunde met computeralgebra, M. Kamminga-van Hulsen, P. Gondrie & G. van Alst, (140 toepassingsgerichte opgaven, uitgewerkt in DERIVE and Maple)
Academic Service, Schoonhoven, NL, 1994, ISBN 90 6233 956 3

One of the results of the German DUG Meeting in Nuremberg was the idea to establish an exchange for DERIVE teaching materials in the DNL. If you have materials to offer then please write, fax or call. I will announce it in the DNL, and if there are other members who are interested in the papers or diskettes they should organize contact with each others.

I can offer: The Binomial Theorem, GCD & LCM, The System of Coordinates (in English and in German).

Modellversuch

**Entwicklung und Erprobung von Materialien
zum Einsatz von Computeralgebra Systemen
im Mathematikunterricht der Sekundarstufe II**

Das jeweilige Computeralgebra System soll – wie heute der gewöhnliche Taschenrechner – als ständig präsentes Werkzeug beim Mathematik Lernen Verwendung finden. Die Materialien (Medienpakete aus Buch & Disketten) werden die Funktion eines vollständigen Mathematik-Lehrwerks übernehmen, d.h. den gesamten Mathematikunterricht neuer Form unterstützen. Sie sollen ferner innovativ auf die Lehrplanentwicklung wirken.

Solange allerdings die herkömmlichen Lehrpläne noch in Kraft sind, müssen in mannigfacher Weise Kompromisse eingegangen werden.

Am Modellversuch sind Lehrkräfte ausgewählter Celler Gymnasien und insbesondere Mitglieder des Staatlichen Studienseminars Celle sowie weitere Versuchsschulen beteiligt. Der erste Materialienband zur **Analysis 1** zunächst für das Computeralgebra System DERIVE wird zu Beginn des Schuljahres 1995/96 vorliegen (Erprobungsfassung). Er wird später in einem namhaften Schulbuchverlag publiziert werden.

Eine – noch unvollständige – Vorfassung (100 Seiten, Ergebnis der Vorlaufstudien) kann ab sofort von interessierten Einzelpersonen oder Schulen zum Selbstkostenpreis (10.- DM in Briefmarken) bezogen werden.

Kontakt: StD Rüdeger Baumann, Gymnasium Ernestinum, Burgstr. 21, 29221 Celle

Liebe DUG-Mitglieder.

Nach einem arbeitsreichen Frühjahr freuen wir uns alle auf den verdienten Urlaub. Wie Sie aus früheren Informationen wissen, haben um die Osterzeit in Deutschland zwei wichtige Veranstaltungen stattgefunden. Die – auch für DERIVE sehr erfolgreiche – MNU-Hauptversammlung in Nürnberg gab den Rahmen für das 2. Deutsche DUG- Treffen. Herzlichen Dank allen Besuchern des interessanten gemeinsamen Nachmittags. Ein Ergebnis ist die Einrichtung einer DERIVE-Materialien-Börse im DNL. Näheres finden Sie auf der Informationsseite. Lieber Wolfgang Pröpper herzlichen Dank für die Organisation des Treffens.

Die DERIVE Days Düsseldorf waren von ca. 250 interessierten Lehrern aus Deutschland. Österreich, Holland und England sehr gut besucht. An dieser Stelle herzlichen Dank an Bärbel Barzel und Leo Klingen für die professionelle Abwicklung. Ich habe mich besonders gefreut, so viele von Ihnen wieder oder zum ersten Mal zu treffen.

Auf Seite 26 finden Sie zwei Produkte aus meinem Grafik-Workshop. Einer der nächsten DNLs wird hauptsächlich der Geometrie gewidmet sein.

Das nächste große Ereignis wird die DERIVE-Konferenz in Hawaii im August sein, zu der sich 37 Teilnehmer aus 12 Nationen (darunter die VAR und Malaysia) angemeldet haben. Ich werde im Herbst darüber berichten.

Sie finden in diesem Heft einen Beitrag mit etwas Höherer Mathematik über die Besselfunktionen. Aus Platzmangel muss dieses Mal auf die nächste Folge des Kurvenlexikons verzichtet werden, aber die Kardiode und ein Beitrag von Peter Baum zum Kurvenlexikon liegen schon bereit. 40 Seiten Umfang sind offensichtlich noch zu wenig.

Wir legen dieser Nummer eine Quittung für Ihren Mitgliedsbeitrag für 1995 bei. Bitte verwechseln Sie das nicht mit einer Rechnung oder Zahlungsaufforderung - außer es steht "ERINNERUNG" darauf

Abschließend wünsche ich Ihnen einen schönen Sommer und freue mich schon auf die Zusammenstellung des nächsten DNL.

Ihr Josef Böhm

Dear DUG-Members,

After a busy spring we are looking forward to well-deserved holidays. As you will know from earlier information two important events took place in Germany at Easter time. The MNU-meeting in Nuremberg - very successful for DERIVE - was the frame for the 2nd German DUG-meeting. Many thanks to all participants for joining an interesting afternoon. One result is the establishment of a DERIVE-Materials-Exchange in the DNL. (See the first page please). Dear Wolfgang Pröpper, many thanks for making the meeting possible and successful.

About 250 teachers from Germany, Austria, Holland and England attended the DERIVE Days Düsseldorf. Many thanks from this place to Bärbel Barzel and Leo Klingen for their professional organization. I had the pleasure to meet there so many of you again or the first time.

On page 26 you will find two products from my Graphic-Workshop. One of the next DNLs will be dedicated mainly to geometric items.

The next big event will be the Hawaii DERIVE Conference in August. 37 ladies and gentlemen from 12 nations (UAR and Malaysia among them) have registered. I will give a report in the next DNL.

In this issue you will find one contribution of a higher mathematical level about Bessel functions. You will miss the Lexicon of Curves. But the Cardiode and Peter Baum's ideas inspired by the Lexicon are ready for the next DNL. Believe it or not, 40 pages seem to be too few.

We enclose a receipt for the 1995 *membership dues. Please don't misunderstand this receipt as an invoice, except you find "REMINDER" on it.*

At the end of my letter I wish you a fine summer and I am looking forward to producing the next DNL for you.

Sincerely yours

Josef Böhm

The *DERIVE-NEWSLETTER* is the Bulletin of the *DERIVE User Group*. It is published at least four times a year with a contents of 30 pages minimum. The goals of the *D-N-L* are to enable the exchange of experiences made with *DERIVE* as well as to create a group to discuss the possibilities of new methodical and didactical manners in teaching mathematics.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone: 43-(0)2275/8207

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 *D-N-L*. It must be said, though, that non-English articles are very welcome nonetheless. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in the *DERIVE Newsletter*. The more contributions you will send, the more lively and richer in contents the *DERIVE Newsletter* will be.

Preview: (Contributions for the next issues):

Stability of systems of ODEs, Kozubik, SLO
Prime Iterating Number Generators, Wild, UK
Graphic Integration, Probability Theory, Linear Programming, Böhm, A
LOGO in DERIVE, Lechner, A
DREIECK.MTH, Wadsack, A
IMP Logo and Misguided Missiles, Sawada, HAWAII
3D Geometry, Reichel, A
Parallel- and Central Projection, Böhm, AUS
Conic Sections and their 3D Visualisation, Fuchs & Böhm, A
Müller´s Method to solve univariate equations a.o., Speck, NZL
Vector and Vector Indices Sorting, Biryukov, RUS
Clear Function Parameters Representation, Biryukov, RUS
Algebra at A-Level, Goldstein, UK
Tilgung fremderregter Schwingungen, Klingen, GER
Utility for Complex Dynamic Systems, Lechner, A
Some Improvements on the Resolution of ODEs, Fuster, E
Lexicon of Curves (7) – the Cardiod, Weth, GER
Bézier Curves with DERIVE, Scheu, GER
and
Setif, FRA; Vermeylen, Belgium; Lymer, FRA; Leinbach, USA, Aue, GER;
Wiesenbauer, A; Keunecke, GER; Roeloffs, NL; Baum, GER

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

G P Speck, Wanganui, New Zealand

I have received DNL#17 which contains several articles of interest to me. The DERIVE Newsletters certainly serve a useful purpose.

Though elementary, the issue of solving a simultaneous system of linear equations with a number of characters too great to Author (edit) all at once seems to bother virtually all students when first encountered. Thus, if this matter has not been raised in previous NEWSLETTERS, I offer the enclosed note.

The following "real life" example may be of interest to DUG readers.

In considering VENN diagrams showing probabilities for events, 32 simultaneous linear equations in 32 unknown arise in a natural way in considering a Venn diagram for 5 (completely) independent events. Such a system of equations is given in the DERIVE printout at the end of this discussion. The number of characters is too great to Author (Enter) all at once. However, as usually the case with DERIVE there are ways to circumvent a difficulty. In this case one approach is to use APPEND after entering the system of equations piecemeal over several lines.

(I don´t give a copy of the printout because it´s too big. I try to explain, editor.)

Name the unknown x01, x02, …, x32. This provides an ordering from the 1^{st} through the 32^{nd} final solution matrix. Then enter in 4 lines 4 times 8 equations, giving (Set Input Mode Word):

```
#1:  [x24 + x19 + …]
#2:  [x11 + x12 + …]
#3:  [x20 + x25 + …]
#4:  [x16 + x22 + …]
#5:  [#1, #2, #3, #4]
```

Then Simplify followed by Solving the system.

Kurt Schmidt, Köln, Germany

Gestern erhielt ich das Handbuch für DERIVE 3. Es ist natürlich voller Fehler. Auf Seite 146 ist die Formel für r falsch.

Als es noch keinen Computer gab und man alles mit dem Rechenschieber berechnen musste, konnte man natürlich dieses Problem nicht so elegant behandeln. Wir machten es damals höchst ungeschickt und setzten $y = a\,x$. Man erhält dann eine Parameterdarstellung:

K. Schmidt points out that the manual of DERIVE 3 contains some mistakes. One of them can be found on page 151. The formula for r is wrong. In earlier times – without computer support – they did it very clumsy and set $y = a\,x$, which led to a parameter representation:

$$x = \frac{t^4 - 1}{1 - t^5} \rightarrow \left[\frac{t^4 - 1}{1 - t^5}, t \cdot \frac{t^4 - 1}{1 - t^5}\right] \quad (-10 \le a \le 10).$$

The given problem is: Implicit plot of $x^5 + x^4 = y^5 + y^4$. Substituting $r \cos\theta$ for x and $r \sin\theta$ for y, then solving for r gives two explicit solutions:

$$r = \frac{1 - 2\cos^2\theta}{\cos^5\theta - \sin^5\theta} \quad \text{or} \quad r = \frac{-\cos 2\theta}{\cos^5\theta - \sin^5\theta}.$$

See the DERIVE realisation (version 6.1) in the following:

#1: $x^5 + x^4 = y^5 + y^4$

#2: $r^5 \cdot \text{COS}(\theta)^5 + r^4 \cdot \text{COS}(\theta)^4 = r^5 \cdot \text{SIN}(\theta)^5 + r^4 \cdot \text{SIN}(\theta)^4$

#3: $\text{SOLVE}(r^5 \cdot \text{COS}(\theta)^5 + r^4 \cdot \text{COS}(\theta)^4 = r^5 \cdot \text{SIN}(\theta)^5 + r^4 \cdot \text{SIN}(\theta)^4, r)$

#4: $r = \dfrac{1 - 2\cdot\text{COS}(\theta)^2}{\text{COS}(\theta)^5 - \text{SIN}(\theta)^5} \vee r = 0$

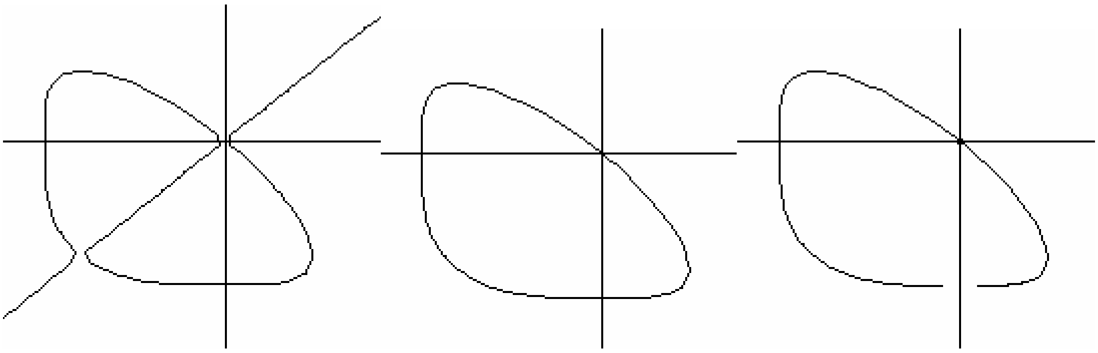
#5: Trigonometry := Collect

#6: Trigpower := Cosines

#7: $r = \dfrac{-\text{COS}(2\cdot\theta)}{\text{COS}(\theta)^5 - \text{SIN}(\theta)^5} \vee r = 0$

#8: $\text{SOLUTIONS}(x^5 + x^4 = y^5 + y^4 \wedge y = t\cdot x, [x, y])$

#9: $\begin{bmatrix} 0 & 0 \\ -\dfrac{t^3 + t^2 + t + 1}{t^4 + t^3 + t^2 + t + 1} & -\dfrac{t\cdot(t^3 + t^2 + t + 1)}{t^4 + t^3 + t^2 + t + 1} \end{bmatrix}$



implicit plot | polar plot | parametric plot

Comparing the quality of the plot shows that
(1) only the implicit plot presents the solution $y = x$ and
(2) the best quality plot of the curve gives the polar form.

Additional comment from Kurt Schmidt:

The formula on page 21 in revised DNL#11 (Digital Filters Design) can be simplified. Instead of:

`TRIW(k,n) := (ABS(k+n+1)/2+ABS(k−n−1)/2−ABS(k)) WR(k,n)/(n+1)` one can write:

`TRIW(k,n) := (n+1−ABS(k)) WR(k,n)/(n+1).`

Pam Bishop sent some messages from the email DERIVE News Group which is established at the Birmingham University. Many thanks, Pam, and the best to you.

Here is a question from Alfonso Poblacion together with A. van der Meer´s answer:

If we try to calculate

```
INT(COS(SQRT(x)),x)
```

DERIVE gives us

```
2*COS(SQRT(x))+2*SQRT(x)*SIN(SQRT(x))
```

that is right.

Then we want to find the area of the region between the graph of y = cos(sqrt(x)) and the x-axis in [0, $4\pi^2$]. If we try with

```
INT(COS(SQRT(x)),x,0,4*pi^2)

0
```

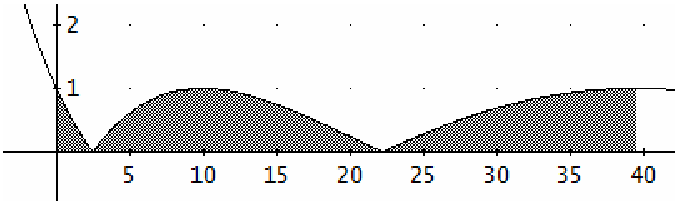
the answer is 0, that is also correct, But if we try

```
INT(ABS(COS(SQRT(x))),x,0,4*pi^2)

0
```

the answer is still 0, and this is absurd. If you try with many other functions which intersect the x-axis, like sin(x), cos(x), etc. the answers are perfect. What happens with this function? Maybe there will be others.

Later versions of DERIVE give the correct result:

$$\int_0^{4\cdot\pi^2} |\cos(\sqrt{x})|\, dx = 8\cdot\pi$$



A. van der Meer´s comment was very interesting in 1995. I reactivated DERIVE (version 3.14) and reproduced his answer:

One has to be very cautious in integrating values. DERIVE is clever, but it is only a machine. Let us see what happens:

#1: $\int_0^{4\cdot\pi^2} |\cos(\sqrt{x})|\, dx = 0$

#2: gives the wrong answer. Without the absolute value, the primitive is correct.

#3: $\int \cos(\sqrt{x})\, dx$

#4: $2\cdot\cos(\sqrt{x}) + 2\cdot\sqrt{x}\cdot\sin(\sqrt{x})$

#5: But:

#6: $F(x) := \int |COS(\sqrt{x})|\, dx$

#7: simplifies to:

#8: $SIGN(COS(\sqrt{x}))\cdot(2\cdot COS(\sqrt{x}) + 2\cdot\sqrt{x}\cdot SIN(\sqrt{x}))$

#9: which has a discontinuity at x = π^2, due to the SIGN-function that jumps from +1 to -1:

#10: $\lim_{x\to(\pi/2)^2-} F(x)$

#11: π

#12: $\lim_{x\to(\pi/2)^2+} F(x)$

#13: $-\pi$

#14: If we start at (π/2)^2 a rather remarkable primitive is calculated:

#15: $\int_{\pi^2/4}^{x} |COS(\sqrt{t})|\, dt$

#16: $\pi\cdot SIGN(0) + SIGN(COS(\sqrt{x}))\cdot(2\cdot COS(\sqrt{x}) + 2\cdot\sqrt{x}\cdot SIN(\sqrt{x}))$

#17: But this non-existing π·SIGN(0) cancels if a definite integral is calculated:

#18: $\int_{\pi^2/4}^{\pi^2} |COS(\sqrt{t})|\, dt$

#19: $\pi + 2$

#20: Similar problems can arise if a substitution is made which is not continuous on the integration interval.

#23: $$F(x) := \int_0^x \frac{1}{2 - SIN(t)}\, dt$$

#24: $$\frac{2\cdot\sqrt{3}\cdot ATAN\left(\frac{COS(x)}{SIN(x) - \sqrt{3} - 2}\right)}{3} + \frac{\sqrt{3}\cdot\pi}{18} + \frac{\sqrt{3}\cdot x}{3}$$

#25: $F(2\cdot\pi)$

#26: $$\frac{2\cdot\sqrt{3}\cdot\pi}{3}$$

#24 is a correct answer (for example F(2π)): DERIVE avoids the "tan(t/2)"-mistake successfully. But, doing the substitution "by hand" (first load Utility MISC.MTH)

#27: $H(t) := \text{INT\_SUBST}\left(\frac{1}{2 - \text{SIN}(t)},\ t,\ \text{TAN}\left(\frac{t}{2}\right)\right)$

#28: $$-\frac{2\cdot\sqrt{3}\cdot\text{ATAN}\left(\frac{\sqrt{3}\cdot(\text{COS}(t) - 2\cdot\text{SIN}(t) + 1)}{3\cdot(\text{COS}(t) + 1)}\right)}{3}$$

#29: $$H(t) := -\frac{2\cdot\sqrt{3}\cdot\text{ATAN}\left(\frac{\sqrt{3}\cdot(\text{COS}(t) - 2\cdot\text{SIN}(t) + 1)}{3\cdot(\text{COS}(t) + 1)}\right)}{3}$$

#30: $H(\pi)$

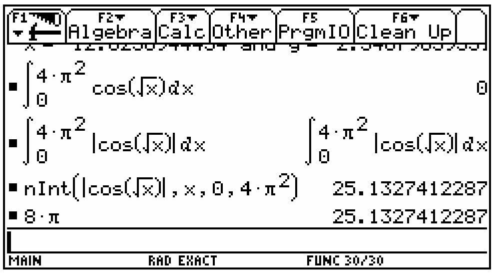
#31: ?

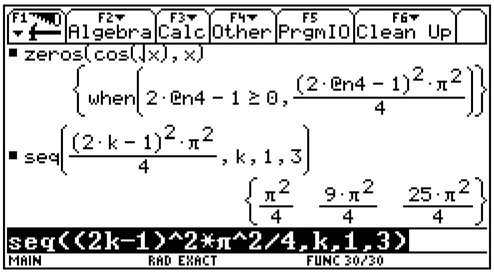
#32: $H(2\cdot\pi) - H(0)$

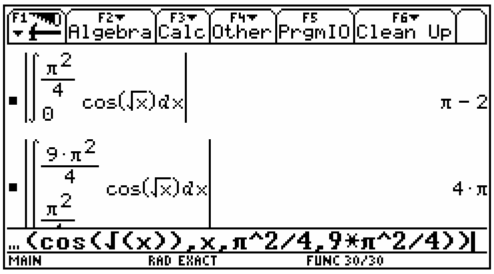
#33: 0

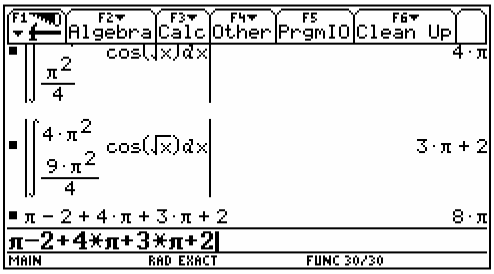
H(t) has a discontinuity at t = π. And of course #32 gives the wrong answer 0.

How do the TI-calculators perform?









The next messages are from the Bulletin Board Service. As I´ve heard in many talks when I was round in Europe many of our members appreciate the ideas, the hints and the conversations of the BBS.

Message 3392: From KEITH WILLIAMS to HADUD about DEG TO HMS AND HMS TO DEG

Dear Hadud, I came up with functions which might be good to add to your DERIVE functions. You may have already made these functions. They are functions which will convert between degrees to hours, minutes, seconds and hms to degrees. These two functions use two other functions which extract the integer part of a number and the fractional part of a number.

```
ip(m):=IF(m<0,-FLOOR(ABS(m)),FLOOR(m))

fp(m):=IF(m<0,-MOD(ABS(m)),MOD(m))

deg_2_hms(y):=ip(fp(y)*60)/100+ip(y)+fp(fp(y)*60)*60/10000

hms_2_deg(y):=ip(y)+ip(fp(y)*100)/60+100*fp(100*y)/3600
```

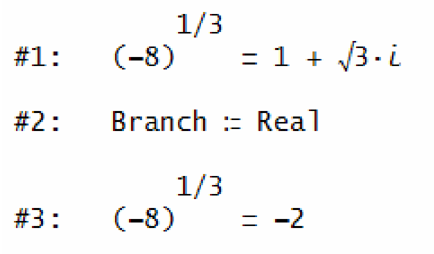
Maybe these functions will better improve DERIVE since DERIVE does not have these functions. Keith Williams

Message 3398: From MONA to PUBLIC about INVERSE FUNCTIONS

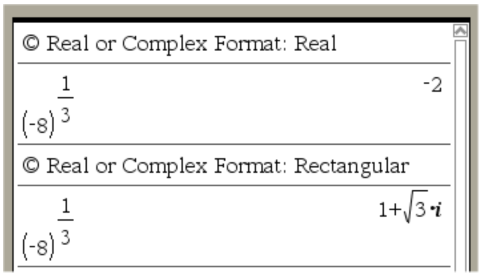
When I graph the inverse function of $f(x) = x^3 - 1$, I only get the portion where $x \geq 0$.
Also, if I try to evaluate a function such as the inverse of $f(x)$ which involves a fractional exponent such as 2/3 which should yield real values for negative values of x, I get complex answers.
Note: On my TI-81 calculator I solved this problem by rewriting x^(2/3) as (x^(1/3))^2 but DERIVE would not Simplify this. Any help would be appreciated.

Message 3399: From JERRY GLYNN to MONA about #3398 INVERSE FUNCTIONS

Derive is complex number oriented. Try (–8)^(1/3) and Simplify and you´ll get a complex number and not –2. –2 is $-2 + 0 \cdot i$ and would be plotted on the complex plane off to the left at 180 degrees. Derive´s answer is $1 + \sqrt{3} \cdot i$ which plots in the first quadrant so qualifies as the *principal value*.
Your question is the same as mine when I first used Derive (or Mathematica) and Derive has implemented at my suggestion a bit of flexibility to satisfy us. Do Manage Real (Options > Mode Settings > Branch > Real in Derive 6.10) and then simplify (–8)^(1/3) again and you will get –2 and your plots will give what you want. Keep asking questions.



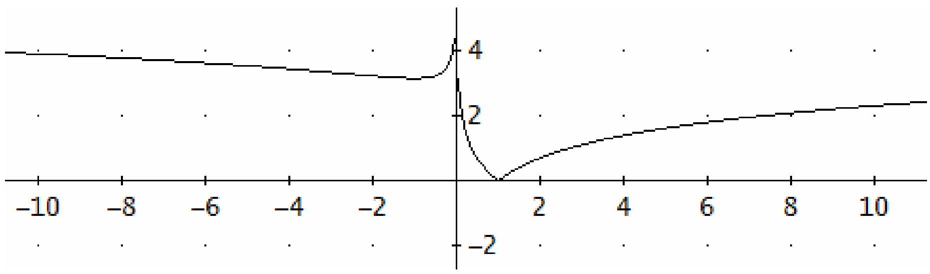
Derive 6.10



TI-Nspire (same as TI-92 and TI-Voyage 200)

Message 3400: From BOOM-BOOM to MONA about #3398 INVERSE FUNCTIONS

Try (x^2)^(1/3) for a TI-81 "look alike" graph of x^(2/3). The real problem (no pun intended) is the complex domain of DERIVE vs the real domain of TI-81.
Another interesting result is to graph y = abs(ln x). DERIVE and the TI-85 will plot something in the second quadrant; TI-81 does not. What´s right? (The absolute value – better known as modulus – of a complex number is a real number.) Adios, Larry Gilligan



Message 3402: From DPHILLIPS to PUBLIC about LARGEST KNOWN PRIME NUMBER

I was able to compute 2^859433 – 1 with Derive XM. Derive could not compute the number directly, perhaps because I have only 4 Megs of RAM so I computed it indirectly. I first calculated 2^200000, multiplied the result by itself, and then multiplied that result by itself to obtain 2^800000. I then multiplied that result by 2^59433 and subtracted 1. The number took 6 minutes and 21.1 seconds to compute and half an hour and 137 pages to print out. It had the required number of digits, 258716 and the number ended in 1. I just hope that Derive was able to compute all of the digits correctly! I gave the printout to my daughter, who is in the 11th grade, and is required to keep a math journal. I hope her teacher and class appreciate that the 137 page print out is the largest known prime number.
Don Phillips

Message 3403: From VICTOR SANTOS to PUBLIC about INVERSE FUNCTIONS

After reading Jerry Glynn´s reply to Mona about the way Derive handles complex branches, I remembered I found a Derive bug concerning this same subject. I am using Derive 2.55 so I think this bug isn´t corrected in Derive yet.
Everyone knows that (–2)^√3 is ALWAYS COMPLEX, no matter what branch you choose. With Derive when you Approximate this expression you get the correct answer if Branch is Principal (you get a complex number), but when you choose Branch Real you get an erroneous reply (a real number). It´s interesting to note that you get correct answers in both cases if you SIMPLIFY the expressions instead of using APPROX. Obviously there is something wrong here because the answer should always be complex.

```
Branch:=Real

(-2)^SQRT(3)

;Approx(#1)
2.21288-2.47766*#i

Branch:=Real

;Approx(#1)
3.32199
```

Should someone from the Derive team see this, I would like to know the reason for this inconsistency.
Victor Santos, from Portugal, in Europe.

Message 3404: From HARALD LANG to VICTOR SANTOS about #3403

No, I don´t agree that this is a bug in DERIVE. On the contrary, it is an example on how carefully DERIVE is designed. The reason is the following: $(-2)^q$, where q is an irrational number, has infinitely many values, and they are distributed as a dense subset of the circle of radius 2^q. I.e., every point on that circle has a value of $(-2)^q$ arbitrary close to it; in other words, any point on that circle is an arbitrary approximation to $(-2)^q$. When DERIVE approximates, using the real branch, she chooses 2^q, which is real, and an approximation to any degree of accuracy. IMHO, that is the most relevant choice.
--- Harald

Message 3406: From VICTOR SANTOS to HARALD LANG about #3405

Thanks for the explanation about the way Derive handles infinitely many values of $(-2)^q$, q being irrational, when we approximate using the Real Branch. It now seems reasonable to me to get a real value for APPROX of $(-2)^{\sqrt{3}}$.

Nevertheless, something still is not quite clear to me and I again ask for your comment on the following: $(2 + i)^{\sqrt{3}}$ is ALWAYS COMPLEX, too and its infinitely many values all lie on the circle of radius 5^(√3/2). Following your explanation, there is a real approximation to any degree of accuracy to $(2 + i)^{\sqrt{3}}$ and that is 5^(√3/2). But in this case Derive will not APPROXimate to this real value, even when we choose the Real Branch.
I would appreciate your comments on this. Thanks again, Victor Santos.

Message 3408: From HARALD LANG to VICTOR SANTOS about #3406

Hmm, no I can´t see any good reason for this. I suggest you explicitly address your question to Soft Warehouse, so they get alerted. Or does Jerry Glynn (Hi, Jerry, I suppose you read this.) have any suggestion as to why DERIVE behaves like this? --- Harald

#1: $(-2)^{\sqrt{3}}$

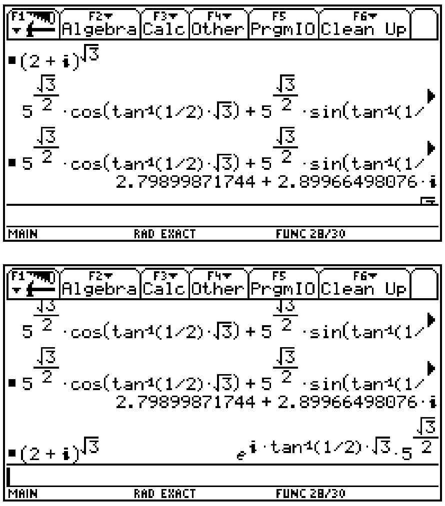
#2: Branch := Principal

#3: $2.212884986 - 2.477661129\cdot i$

#4: Branch := Real

#5: $2.212884986 - 2.477661129\cdot i$

Derive 3.14



TI-V200 and TI-Nspire

#6: $(2 + i)^{\sqrt{3}}$

#7: Branch := Real

approximated

#8: $2.798998717 + 2.899664980\cdot i$

simplified

#9: $5^{\sqrt{3}/2}$

#10: Branch := Principal

#11: $5^{\sqrt{3}/2}\cdot e^{i\cdot(\sqrt{3}\cdot\pi/4 - \sqrt{3}\cdot \mathrm{ATAN}(1/3))}$

#12: $2.798998717 + 2.899664980\cdot i$

Derive 6.10

Compare how the behaviour is changing as time goes by.

Message 3411: From JERRY GLYNN to PUBLIC about FORM OF ANSWERS IN DERIVE

In Derive the derivative of certain expressions is handled correctly but in a form which I believe is a mistake in judgment. What do you think?

#1: $\frac{d}{dx}(x^{25} + (1 + x + x^2)^{117})$

#2: $234\cdot x^{233} + 27261\cdot x^{232} + 1601496\cdot x^{231} + 63225162\cdot x^{230} + 1886202630\cdot x^{229} + 45338551947\cdot x^{228} +$
$914299617348\cdot x^{227} + 15904855609578\cdot x^{226} + 243557328977586\cdot x^{225} + 3334319272823175\cdot x^{224} +$

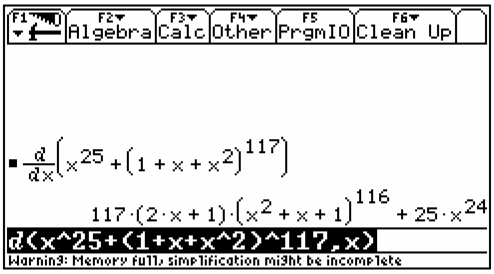
If I did this expression by hand I never expand the polynomial but Derive does.
My answer is $25x^{24} + 117(1 + x + x^2)^{116}\,(1 + 2x)$ and I am done. I think that Derive should not expand when asked to differentiate. I am sure there are other sides to this question which I am not seeing. Comments? Is this part of a larger question which we might discuss?

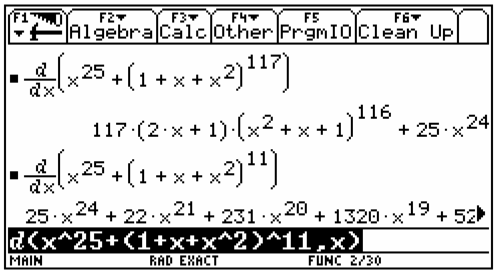
Message 3412: From HARALD LANG to JERRY GLYNN about #3411

Hi Jerry,
I am glad you bring this up, since I have had similar problems, and I have also posted messages about it here, but never got any responses. I have also asked SWHH if they could explain a little how DERIVE "thinks", so one could detect what she is trying to do in similar situations and figure out a way to prevent her (but no answer, alas).
For example: if you split the expression and differentiate x^{25} and $(1+x+x^2)^{117}$ separately, then she is doing ok, which you have no doubt discovered. Another way out, which is the way I currently use in similar situations, is to replace "117" by, say "a", where "a" is declared a positive number. I.e., I ask DERIVE to differentiate a more general function. She will now come up with the expected answer, with a term $(\ldots)^{a-1}$. Now, after you have substituted back 117 for a, you must not simplify the whole expression, since then she starts expanding again; rather you must highlight only the exponent "117–1" (from a–1 above) and simplify that to get the answer we want.

I think it is extremely important to know that such things happen, and that there are ways out. I.e., when you try to do some calculations, and DERIVE just works and works and eventually runs out of memory, you should not give up. It seems to me that DERIVE is sometimes "over-ambitious" as to simplifications, and then the trick is to prevent her from even trying, by making it impossible. In this case she cannot expand $(\ldots)^a$, since a is not a number, as opposed to $(\ldots)^{117}$ where she tries, until she runs out of memory (I don't have the XM-version.) --- Harald

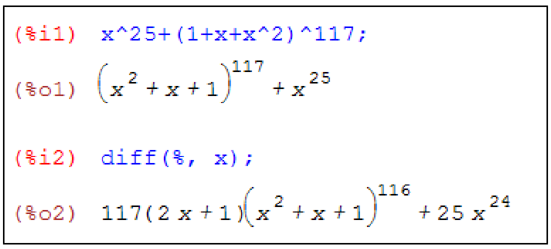
See how other CAS behave in this case (btw, DERIVE has not changed its behaviour since 1995!)





As you can see on the right screen shot the TIs (V200 and NSpire as well) also try to expand before finding the derivative – or try to expand the derivative – but in case of "too much" calculating and displaying the result for the handheld´s memory, both give up and display the unexpanded result – exact what we want the device to do.

Now let´s look how MAXIMA is displaying the derivative (*MATHEMATICA* behaves pretty the same!):



Message 3413: From JERRY GLYNN to HARALD LANG about #3412

Thank you for your usual thoughtful answer. Somehow, I think all of this is related to the following situation: if I have *function*(*something*) I may want to do the *something* first and then do *function* of that result OR I may do better to do *function* of *something* first and then either carry out the *something* or not. I suppose it gets out of hand if a number of these are nested so I'll ignore that for the moment. If I understand SWHH correctly (I often don't) they believe that Simplify should automatically do the right thing and then DERIVE will continue to deserve the ease of use reputation which it has earned.
I believe this view needs modifying since in some cases it is impossible for DERIVE to know what is best. I, in fact, for my own reasons might like to carry out a calculation in more than one way. My only solution is for DERIVE to give the user some control of the situation.

Maybe two versions of *Simplify*:
Insimplify and *Outsimplify* for inside first or outside first. One of the great strengths of DERIVE is its appearance of simplicity. If my suggestion were accepted this could be in danger so I understand if SWHH is slow to make such a change. Also, it may just be a bad idea.
Thanks for your contributions to this forum … I'm always interested in your comments.

Message 3420: From ANDROMEDA to PUBLIC about AVERAGES

A silly math problem has bugging me for some time now which may be simple for most of you out there. Text defines the AVERAGE speed as Total Distance/Total Time (a slope). At the same time it describes AVERAGE value as $(a1+a2+a3+ \dots)/n$ or $(1/(b-a)\cdot\int f(x)\, dx)$. For example:

A	B	C(=A/B)
2.40	5.10	0.47
1.30	3.80	0.34
9.10	15.00	0.60
5.90	6.30	0.94
5.00	6.30	0.79

Total A/Total B = 0.65, while Average of column C = 0.63. Both averages have a different value, the question is which one of these is valid and WHY? Anyone?

Message 3421: From JERRY GLYNN to ANDROMEDA about #3420 / AVERAGES

I say 0.65 is correct. This is the sum of the A's divided by the sum of the B's. Suppose A is the miles you travel and B is the number of minutes to do this travel. Adding up the distance and dividing by the time added up gives you the average speed for this time in miles per minute. Why doesn't the second procedure work? You will get the right answer by the second procedure IF each time segment is the same. That way you are averaging speeds which occur for equal times. Otherwise your average favours those with a small time. A good trick is to move to extreme cases. If you go 500 miles per hour for 23 hours and 10 miles per hour for one hour will your average be (500+10)/24? If you travel 12 hours at 500 mph and 12 hours at 10 mph then your average would be (500+10)/24.

Any help? Keep asking questions and you will learn.

Message 3423: From VICTOR SANTOS to JERRY GLYNN about #3421 / AVERAGES

The average problem here is simple; when averaging speeds you use the harmonic mean and not the arithmetic mean. Harmonic mean can be calculated as $1/h = (1/a1+1/a2+\ldots)/n$.

$$1/h=\frac{\frac{a1}{c1}+\frac{a2}{c2}+\frac{a3}{c3}+\frac{a4}{c4}+\frac{a5}{c5}}{a1+a2+a3+a4+a4}=\frac{\frac{2.4}{0.47}+\frac{1.30}{0.34}+\frac{9.10}{0.60}+\frac{5.90}{0.94}+\frac{5.00}{0.79}}{23.70}=\frac{b1+b2+b3+b4+b5}{23.70}$$

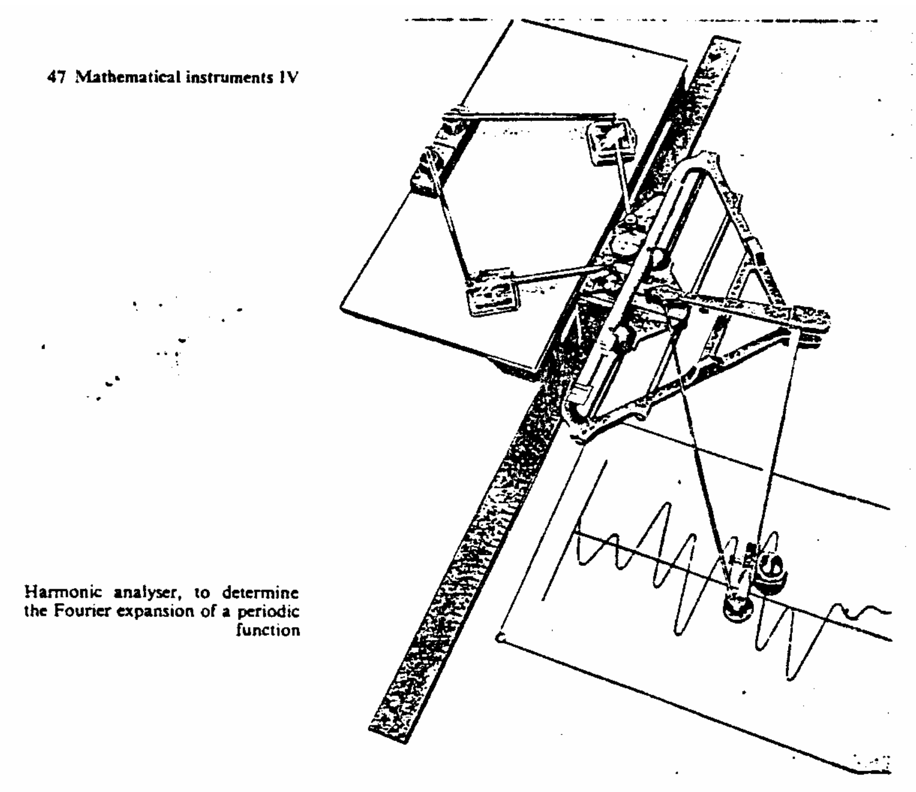
$$h=\frac{23.70}{36.50}\approx 0.6493$$

Concerning "Harmonic":

Daniel J. O'Connel
1219 w. Russel
SAN ANTONIO, Texas 78201

sent a letter with a nice picture of a mathematical instrument, a

Harmonic Analyser.



Harmonic analyser, to determine the Fourier expansion of a periodic function

He asks:

Can you find who makes this in Austria or Germany?

I am trying to help Daniel. Is there anybody in the big DERIVE family who knows an answer?

AN ILLUSTRATION OF THE CONTINUED FRACTIONS METHOD IN DERIVE: THE EVALUATION OF BESSEL FUNCTIONS OF FRACTIONAL AND INTEGER ORDER

J. Cerdán, P. Fernández de Córdoba, D. Ginestar and Yu. L. Ratis

1 Introduction

In this paper a double objective is presented in the framework of the Derive system and its tutorial utilization. In section 2 we insist in how the continued fractions representation of a given function can be used as an approximation of this function, that generally converges faster than another approximation as the Taylor expansion. In sections 3 and 4, we introduce an algorithm to calculate Bessel functions (BFs), of integer and fractional orders, based on the continued fractions method. This proposed algorithm is very efficient numerically [1]. In fact, the usual numerical methods to calculate BFs take into account normalization relations [2]. In this paper we introduce an algorithm and corresponding Derive code to evaluate regular and irregular BFs without any re-calculation thxough nor- malization relations. Furthermore, the method maintains the stability of each recurrence relation, i.e., we use forward recurrence relations for the BFs of the second kind and backward ones [3] for the BFs of the first kind. The algorithm uses forward recurrence relations to generate irregular BFs and takes into account the continued fraction method to evaluate high order regular BFs. From these values we can generate regular BFs applying backward recurrence relations.

The method to evaluate the continued fractions representation of a function and its application to the calculation of BFs has been programmed to be used with Derive, which is an exceptionally easy-to-use computer system with simple structures and a great pedagogical value. The code has been designed in an interactive way and this allows the user to follow the process step by step.

2 Evaluation of continued fractions

In refs [4] and [5] we can find a set of operative definitions about continued fractions.

A continued fraction is given by

$$f(x) = b_0 + \cfrac{a_1}{b_1 + \cfrac{a_2}{b_2 + \cfrac{a_3}{b_3 + \cfrac{a_4}{b_4 + \dots}}}}. \tag{1}$$

Printers prefer to write this expression in the following way

$$f(x) = b_0 + \frac{a_1}{b_1 +}\frac{a_2}{b_2 +}\frac{a_3}{b_3 +}\frac{a_5}{b_4 +}\dots \quad . \tag{2}$$

In the previous expressions the *a's* and *b*'s can themselves be functions of *x*. We consider, as an example [4], the following continued fraction representation of the tangent function

$$\tan(x) = \frac{x}{1-}\frac{x^2}{3-}\frac{x^2}{5-}\frac{x^2}{7-}\dots \ , \tag{3}$$

i.e. for this case

$$a_1 = x, \quad a_j = -x^2; \qquad j \geq 2,$$
$$b_0 = 0, \quad b_j = 2j-1; \qquad j \geq 1. \tag{4}$$

Continued fractions frequently converge much more rapidly than power series expansions, and in a much larger domain (not necessarily including the domain of convergence of the series, however). Sometimes, the continued fraction converges best where the series does worst, although this is not a general rule.

How do you tell how far to go with a continued fraction? Unlike a series, you can just evaluate equation (1) from left to right, stopping when the change is small. Written in the form of (1), the only way to evaluate the continued fraction is from right to left, first (blindly!) guessing how far out to start. This is not the right way [4].

The appropriate way to do this, is to use the following result that relates the continued fractions expansions and rational approximations, and gives a means of evaluating (1) or (2) from left to right. Let f_n denote the result of evaluating (2) with coefficients through a_n and b_n.

Then [4]

$$f_n = \frac{\alpha_n}{\beta_n}, \tag{5}$$

where α_n and β_n are given by the follwing recurrence

$$\alpha_{-1} \equiv 1, \ \beta_{-1} \equiv 0,$$
$$\alpha_0 \equiv b_0, \ \beta_0 \equiv 1, \tag{6}$$

and

$$\alpha_j = b_j \alpha_{j-1} + a_j \alpha_{j-2}; \qquad j = 1, 2, \ldots, n,$$
$$\beta_j = b_j \beta_{j-1} + a_j \beta_{j-2}; \qquad j = 1, 2, \ldots, n. \tag{7}$$

In the next sections, we apply this method to evaluate Bessel functions of fractional and integer order.

3 Bessel functions of fractional order

We are interested in presenting a code in Derive to generate the spherical Bessel functions (SBFs) of the first and second kinds. (We restrict our attention to real values of the argument z).

We use the standard Abramowitz and Stegun [5] notation and we introduce the SBFs of the first kind, $j_n(z) = \sqrt{\pi / 2z}\, J_{n+1/2}(z)$, and the SBFs of the second kind, $y_n(z) = \sqrt{\pi / 2z}\, Y_{n+1/2}(z)$, as particular solutions of the differential equation

$$z^2 w''(z) + 2zw'(z) + [z^2 - n(n+1)]\, w(z) = 0; \quad (n = 0, \pm 1, \pm 2, \ldots).$$

In the code we calculate the SBFs of all orders below *Nmax*, i.e. we generate the set

$$SB(z) = \{j_n(z), y_n(z);\ n = 0, 1, 2, \ldots, Nmax\}.$$

For this, we use an algorithm organized according the following steps:

- Evaluate all the SBFs of the second kind, $\{y_n(z);\ n = 0, 1, 2, \ldots, Nmax\}$, taking into account the known values of $y_0(z) = -\cos(z)/z$ and $y_1(z) = -\sin(z)/z - \cos(z)/z^2$, and using the forward recurrence relation

$$y_{n+1}(z) = \frac{2n+1}{z} y_n(z) - y_{n-1}(z) \qquad n = 1, 2, \ldots \tag{8}$$

- Use the continued fractions method [5] to evaluate the ratio

$$\begin{aligned} H(z) &\equiv \frac{j_{Nmax}(z)}{j_{Nmax-1}(z)} = \frac{j_{Nmax+1/2}(z)}{j_{Nmax-1/2}(z)} = \\ &= \frac{1}{2\left(Nmax+\frac{1}{2}\right)z^{-1} -}\, \frac{1}{2\left(Nmax+\frac{3}{2}\right)z^{-1} -}\, \frac{1}{2\left(Nmax+\frac{5}{2}\right)z^{-1} -} \cdots \ . \end{aligned} \tag{9}$$

- Calculate the upper order SBFs of the first kind, $j_{Nmax}(z)$, using the already known values $y_{Nmax}(z)$ and $y_{Nmax-1}(z)$, the ratio $H(z)$, and the value of the Wronskian of SBFs [5]

$$W\left\{j_{Nmax}(z), y_{Nmax}(z)\right\} \equiv j_{Nmax}(z) y_{Nmax-1}(z) - j_{Nmax-1}(z) y_{Nmax}(z) = z^{-2}.$$

Using the previous expression, we can write

$$j_{Nmax-1}(z) = \frac{1}{z^2\left(H(z) y_{Nmax-1}(z) - y_{Nmax}(z)\right)}, \tag{10}$$

and then,

$$j_{Nmax}(z) = H(z) j_{Nmax-1}(z) . \tag{11}$$

Notice that we have calculated not only $j_{Nmax}(z)$ but also $j_{Nmax-1}(z)$.

- Evaluate all the SBFs of the first kind, $\{j_n(z),\ n = 0, 1,2, \ldots, Nmax\}$, taking into account the calculated values of $j_{Nmax}(z)$ and $j_{Nmax-1}(z)$, and using the backward recurrence relation:

$$j_{n-1}(z) = \frac{2n+1}{z} j_n(z) - j_{n+1}(z) . \tag{12}$$

4 Bessel functions of integer order

We extend the algorithm proposed in the previous section, to generate Bessel functions (BFs) of the first and second kind, restricting our attention to real values of the argument z.

We introduce the BFs of the first kind, $J_n(z)$, and the BFs of the second kind, $Y_n(z)$, as particular solutions of the differential equation

$$z^2 w''(z) + z w'(z) + \left(z^2 - n^2\right) w(z) = 0 .$$

In the code, we calculate the BFs of all orders below $Nmax$, i.e., we generate the set

$$B(z) \equiv \left\{J_n(z), Y_n(z);\quad n = 0,1,2,\ldots,Nmax\right\}.$$

Now, the algorithm is organized in the following way:

- Evaluate $Y_0(z)$ and $Y_1(z)$ using [5]

$$Y_0(z) = \frac{2}{\pi} \ln\left(\frac{z}{2}\right) J_0(z) - \frac{2}{\pi} \sum_{k=0}^{\infty} \frac{\Psi(k+1)(-\frac{1}{4}z^2)^k}{(k!)^2}, \tag{13}$$

and

$$Y_1(z) = -\frac{2}{\pi z} + \frac{2}{\pi}\ln\left(\frac{z}{2}\right)J_1(z) - \frac{z}{2\pi}\sum_{k=0}^{\infty}\left\{\Psi(k+1)+\Psi(k+2)\right\}\frac{\left(-\frac{1}{4}z^2\right)^k}{k!(k+1)!}, \tag{14}$$

where $J_0(z)$ and $J_1(z)$ are the BFs of the first kind of orders 0 and 1, and $\Psi(n)$ is defined by

$$\Psi(n) = -\gamma + \sum_{k=1}^{n-1} k^{-1},$$

with γ being Euler´s constant.

- Evaluate all the BFs of the second kind, $\{Y_n(z), n = 0, 1, 2, \ldots, Nmax\}$, taking into account the calculated values of $Y_0(z)$ and $Y_1(z)$ and using the forward recurrence relation

$$Y_{n+1}(z) = \frac{2n}{z}Y_n(z) - Y_{n-1}(z), \quad n = 1, 2, \ldots \ . \tag{15}$$

- Use the continued fractions method to evaluate the ratio

$$H(z) \equiv \frac{J_{Nmax}(z)}{J_{Nmax-1}(z)} = \frac{1}{2Nmax\,z^{-1} -}\;\frac{1}{2(Nmax+1)z^{-1} -}\;\frac{1}{2(Nmax+2)z^{-1} -}\cdots \ . \tag{16}$$

- Caculate the upper order BFs of the first kind, $J_{Nmax}(z)$, using the already known values of $Y_{Nmax}(z)$ and $Y_{Nmax-1}(z)$, the ratio $H(z)$ and the value of the Wronskian of the BFs [5]

$$W\{J_{Nmax-1}(z), Y_{Nmax-1}(z)\} \equiv J_{Nmax}(z)Y_{Nmax-1}(z) - J_{Nmax-1}(z)Y_{Nmax}(z) = \frac{2}{\pi z}.$$

As in the previous section, we can write

$$J_{Nmax}(z) = \frac{2}{\pi z\,(Y_{Nmax-1}(z) - Y_{Nmax}(z)/H(z))}, \tag{17}$$

and then

$$J_{Nmax-1}(z) = \frac{J_{Nmax}(z)}{H(z)}. \tag{18}$$

Notice again that we calculate not only $J_{Nmax}(z)$ but also $J_{Nmax-1}(z)$.

- Evaluate all the BFs of the first kind, $\{J_n(z), n = 0, 1, 2, \ldots, Nmax\}$, taking into account the calculated values of $J_{Nmax}(z)$ but also $J_{Nmax-1}(z)$ and using the backward recurrence relation

$$J_{n-1}(z) = \frac{2n}{z}J_n(z) - J_{n+1}(z). \tag{19}$$

We would like to point out that $J_0(z)$ and $J_1(z)$ can be used as checks of the accuracy of the procedure because both are calculated at the end of the method.

5 Program specification

In this section, we introduce three Derive programs. A first code to evaluate the continued fractions representation of the tangent function, a second one to calculate SB(z) and the last one to evaluate B(z). We pass to detail the structure of these programs:

1. Derive program to evaluate the continued fraction representation of the tangent function (see Appendix A).

The code is organized as follows:

- Define the a's and b's in expression (4). We evaluate the values of a_n with the function `A(N_,z)` and the values of b_n with the `B(N_)` function.

- Define the α's and β's in expression (5) using the recurrence relations (6) and (7). We evaluate the values of α_1 and β_1 with the functions `ALPHA(L_,z)` and `BETA(L_,z)`.

- We store the values of $f_k \equiv \dfrac{\alpha_k}{\beta_k}$, ($k = 0, 1, 2, \ldots, N$) in the `H(N,z)` vector.

(I left all files in the form of 1995 unalterred as much as possible. The first file appears in its "DERIVE for DOS" form. Josef.)

```
"Appendix A"
A(N_,z):=IF(N_<1,0,IF(N_=1,z,-z^2))
B(N_):=IF(N_<=0,0,2*N_-1)
ALPHA(L_,z):=IF(L_=-1,1,IF(L_=0,B(L_),IF(L_>=1,B(L_)*ALPHA(L_-1,z)+
             ALPHA(L_,z)*ALPHA(L_-2,z))))
BETA(L_,z):=IF(L_=-1,0,IF(L_=0,1,IF(L_>=1,B(L_)*BETA(L_-1,z)+
            A(L_,z)*BETA(L_-2,z))))
H(N,z):=VECTOR(ALPHA(k_,z)/BETA(k_,z),k_,0,N)
```

See the results in paragraph 6.

2. Derive program to evaluate the BF(z) (see Appendix B).

The code is organized in the following way:

- Evaluate the set $\{y_n(z), n = 0, 1, 2, ..., Nmax\}$ using the forward recurrence relation (8) and the known values of $y_0(z)$ and $y_1(z)$. We evaluate $y_1(z)$ with the `PASO(L_,z)` function, and we store the set of calculated values in the `YNEW(NMAX,z)` vector. In order to compare our results, we also evaluate this set using the `SPHERICAL_BESSEL_Y(k_,z)` function defined in the standard BESSEL.MTH utility file - wihich is now BesselFunctions.mth - storing the results in the `YOLD(NMAX,z)` vector.

- Use of the continued fraction method to evaluate $H(z)$ of expression (9) which we rewrite in the form

$$H(z) = -\frac{-1}{2\left(Nmax+\frac{1}{2}\right)z^{-1}+}\,\frac{-1}{2\left(Nmax+\frac{3}{2}\right)z^{-1}+}\,\frac{-1}{2\left(Nmax+\frac{5}{2}\right)z^{-1}+}\cdots \quad . \tag{20}$$

and so,

$$a_n = -1; \qquad n \geq 1$$
$$b_0 = 0, \quad b_n = \frac{1}{z}(2Nmax + 2n - 1); \quad n \geq 1 \tag{21}$$

We calculate the values of a_n and b_n of the expression (21) with the `A(N_)` and `B(N_,NMAX,z)` functions, and the values of α_l and β_l with the `ALPHA(L_,NMAX,z)` and `BETA(L_,NMAX,z)` functions.

The last step is to store the values of f_k (k = 0, 1, 2, ..., $N\_$) in the `H(N_,NMAX,z)` vector.

By looking at the elements of the `H(N_,NMAX)` vector we get the value of $H(z)$:

$H(z)$ = `ELEMENT(H(N_,NMAX,z),nlim)`

and `nlim` is defined as the component of the vector `H(N_,NMAX,z)` where the changes are small with respect to the previous components.

- Evaluate $j_{Nmax-1}(z)$ and $j_{Nmax}(z)$. `JTOPM1(NMAX,z)` is a function to calculate $j_{Nmax-1}(z)$ using the equation (10) and evaluate $j_{Nmax}(z)$ using the expression (11) with the function `JTOP(NMAX,z)`.
- Evaluate the set $\{j_n(z),\ n = 0, 1, 2, \ldots, Nmax\}$ using the backward recurrence relation (12). We evaluate $j_l(z)$ with the `PASOJ(L_,z)` function, and we store the set of calculated values in the `JNEW(NMAX,z)` vector. In order to compare our results, we also evaluate this set using the `SPHERICAL_BESSEL_J(k_,z)` function, defined in the standard BESSEL.MTH utility file, storing the results in the `JOLD(NMAX,z)` vector.

```
#1:   LOAD(C:\Programme\TI Education\Derive 6\Math\BesselFunctions.mth)
```

Appendix B - BesselFunctions.mth (earlier BESSEL.MTH) are preloaded
Bessel Functions of fractional order (Spherical Bessel Functions)

```
      PASO(L_, z) :=
        If L_ = 0
           - COS(z)/z
#2:        If L_ = 1
              (- SIN(z) - COS(z)/z)/z
              (2·L_ - 1)/z·PASO(L_ - 1, z) - PASO(L_ - 2, z)

#3:   YNEW(NMAX, z) := VECTOR(PASO(k_, z), k_, 0, NMAX)

#4:   YOLD(nmax, z) := VECTOR(SPHERICAL_BESSEL_Y(k_, z), k_, 0, nmax)
```

Use of the continued fractions method

```
#5:   InputMode := Word

      A(N_) :=
        If N_ ≥ 0
#6:        -1
           0

      B(N_, NMAX, z) :=
        If N_ = 0
           0
#7:        If N_ ≥ 1
              1/z·(2·(NMAX + N_) - 1)
              0

      ALPHA(L_, NMAX, z) :=
        If L_ = -1
           1
#8:        If L_ = 0
              B(L_, NMAX, z)
              If L_ ≥ 1
                 B(L_, NMAX, z)·ALPHA(L_ - 1, NMAX, z) + A(L_)·ALPHA(L_ - 2, NMAX, z)
```

```
      BETA(L_, NMAX, z) :=
        If L_ = -1
           0
#9:        If L_ = 0
              1
              If L_ ≥ 1
                 B(L_, NMAX, z)·BETA(L_ - 1, NMAX, z) + A(L_)·BETA(L_ - 2, NMAX, z)
```

$$\#10:\quad \mathrm{H(N\_,\ NMAX,\ z)} := \mathrm{VECTOR}\left(-\frac{\mathrm{ALPHA(k\_,\ NMAX,\ z)}}{\mathrm{BETA(k\_,\ NMAX,\ z)}},\ \mathrm{k\_},\ 0,\ \mathrm{N\_}\right)$$

Spherical Bessel functions of the first kind

$$\#11:\quad \mathrm{JTOPM1(NMAX,\ z)} := \frac{1}{z^2\cdot\left((\mathrm{H(nlim,\ NMAX,\ z)})_{\mathrm{nlim}}\cdot(\mathrm{YNEW(NMAX,\ z)})_{\mathrm{NMAX}} - (\mathrm{YNEW(NMAX,\ z)})_{\mathrm{NMAX}+1}\right)}$$

$$\#12:\quad \mathrm{JTOP(NMAX,\ z)} := (\mathrm{H(nlim,\ NMAX,\ z)})_{\mathrm{nlim}}\cdot\mathrm{JTOPM1(NMAX,\ z)}$$

```
      PASOJ(L_, NMAX, z) :=
        If L_ > NMAX
           0
           If L_ = NMAX
              JTOP(NMAX, z)
#13:          If L_ = NMAX - 1
                 JTOPM1(NMAX, z)
                 If L_ ≥ 0
                    (2·L_ + 3)/z·PASOJ(L_ + 1, NMAX, z) - PASOJ(L_ + 2, NMAX, z)
                    0

#14:  JNEW(NMAX, z) := VECTOR(PASOJ(j_, NMAX, z), j_, NMAX, 0, -1)

#15:  JOLD(NMAX, z) := VECTOR(SPHERICAL_BESSEL_J(k_, z), k_, NMAX, 0, -1)
```

3. **Derive program to evaluate the B(z)** (see Appendix C).

The code is organized in the same way as in the SBFs case:

- Evaluate the set $\{Y_n(z),\ n = 0, 1, 2, ..., Nmax\}$ calculating $Y_0(z)$ and $Y_1(z)$ and using forward recurrence relation (15).

  To evaluate $Y_0(z)$ and $Y_1(z)$ we use the expressions (13) and (14). These expressions are given in terms of infinite series. To get these values we define the vectors VECTOR_Y0(N_,z) and VECTOR_Y1(N_,z) where we store the different partial sums up to order *N_*. The values $Y_0(z)$ and $Y_1(z)$ are given by the component of these vectors where the changes are small enough.

  We define the values Y0 ≡ $Y_0(z)$ and Y1 ≡ $Y_1(z)$. As in the previous example, we store the set $\{Y_n(z),\ n = 0, 1, 2, ..., Nmax\}$ in the YNEW(NMAX,z) vector.

  In the function PASO(L_,z) we define y_1(z).

- Use of the continued fractions method to evaluate $H(z)$ of expression (16).

- Evaluate $j_{Nmax}(z)$ and $j_{Nmax-1}(z)$. This is done using the expressions (17) and (18).

- Evaluate the set $\{j_n(z);\ n = 1, \ldots, Nmax\}$. This is done by using the relation (19).

(Preloading BesselFunctions.mth is not necessary with recent Derive versions. Derive recognizes all functions provided in the utility files which are collected in the MATH-folder).

```
#1:   LOAD(C:\Programme\TI Education\Derive 6\Math\BesselFunctions.mth)
```

Appendix C
BesselFunctions.mth (earlier Bessel.mth) preloaded

Bessel functions of integer order
Bessel functions of the second kind

```
#2:   InputMode := Word

#3:   J0(z) := BESSEL_J(0, z)

#4:   J1(z) := BESSEL_J(1, z)

#5:   PHI(N_) := -γ + HARMONIC_NUMBER(N_ - 1)
```

#6: $$\text{VECTOR\_Y0(NMAX, z)} := \text{VECTOR}\left(\frac{2}{\pi}\cdot\text{LN}\left(\frac{z}{2}\right)\cdot\text{J0}(z) - \frac{2}{\pi}\cdot\sum_{k\_=0}^{m\_}\frac{\text{PHI}(k\_+1)\cdot\left(-\frac{z^2}{4}\right)^{k\_}}{k\_!^2},\ m\_,\ 1,\ \text{NMAX}\right)$$

#7: $$\text{VECTOR\_Y1(NMAX, z)} := \text{VECTOR}\left(-\frac{2}{\pi\cdot z} + \frac{2}{\pi}\cdot\text{LN}\left(\frac{z}{2}\right)\cdot\text{J1}(z) - \frac{z}{2\cdot\pi}\cdot\sum_{k\_=0}^{m\_}\frac{(\text{PHI}(k\_+1)+\text{PHI}(k\_+2))\cdot\left(-\frac{z^2}{4}\right)^{k\_}}{k\_!\cdot(k\_+1)!},\ m\_,\ 1,\ \text{NMAX}\right)$$

```
      PASO(L_, z) :=
        If L_ = 0
           y0
#8:        If L_ = 1
              y1
              (2·L_ - 2)/z·PASO(L_ - 1, z) - PASO(L_ - 2, z)

#9:   YNEW(NMAX, z) := VECTOR(PASO(k_, z), k_, 0, NMAX)
```

Use of the continued fractions method

```
      A(N_) :=
        If N_ ≥ 0
#10:       -1
           0

      B(N_, NMAX, z) :=
        If N_ = 0
           0
#11:       If N_ ≥ 1
              2/z·(NMAX + N_ - 1)
              0

      ALPHA(L_, NMAX, z) :=
        If L_ = -1
           1
#12:       If L_ = 0
              B(L_, NMAX, z)
              If L_ ≥ 1
                 B(L_, NMAX, z)·ALPHA(L_ - 1, NMAX, z) + A(L_)·ALPHA(L_ - 2, NMAX, z)

      BETA(L_, NMAX, z) :=
        If L_ = -1
           0
#13:       If L_ = 0
              1
              If L_ ≥ 1
                 B(L_, NMAX, z)·BETA(L_ - 1, NMAX, z) + A(L_)·BETA(L_ - 2, NMAX, z)
```

#14: $$\text{H(N\_, NMAX, z)} := \text{VECTOR}\left(-\frac{\text{ALPHA(k\_, NMAX, z)}}{\text{BETA(k\_, NMAX, z)}},\ k\_,\ 0,\ N\_\right)$$

Bessel functions of the first kind:

#15: $$\text{JTOP(NMAX, z)} := \frac{2}{\pi\cdot z\cdot\left(\text{ELEMENT(YNEW(NMAX, z), NMAX)} - \frac{\text{ELEMENT(YNEW(NMAX, z), NMAX + 1)}}{\text{ELEMENT(H(nlim, NMAX, z), nlim)}}\right)}$$

```
                                                      1
#16:  JTOPM1(NMAX, z) := ─────────────────────────────────────·JTOP(NMAX, z)
                          ELEMENT(H(nlim, NMAX, z), nlim)

      PASOJ(L_, NMAX, z) :=
        If L_ > NMAX
           0
           If L_ = NMAX
              JTOP(NMAX, z)
#17:          If L_ = NMAX - 1
                 JTOPM1(NMAX, z)
                 If L_ ≥ 0
                    (2·L_ + 2)/z·PASOJ(L_ + 1, NMAX, z) - PASOJ(L_ + 2, NMAX, z)
                    0

#18:  JNEW(NMAX, z) := VECTOR(PASOJ(j_, NMAX, z), j_, NMAX, 0, -1)

#19:  JOLD(NMAX, z) := VECTOR(BESSEL_J(j_, z), j_, NMAX, 0, -1)
```

6 Results

In this section, we have presented some numerical examples of the accuracy of the method.

1) Results of the program to evaluate tan(z) using a continued fraction representation

In the previous section we present a algorithm to evaluate the continued fraction representation of a given function, and in Appendix A we give as an example, a Derive code to calculate the tangent function with the expression (4). In the code, we store the values of f_n (see eq. (5)) in the `H(N,z)` vector. The results of tan(z) for the case $z=\frac{\pi}{4}$ are the following:

```
H(10,pi/4)

"This the result approximating H(10,pi/4) with DERIVE 6.10"

[0,0.7853982300,0.9886893656,0.9997873697,1,1,1,1,1,1,1]

"This was the result in the original MTH-file in 1995 (6 digits)"

[0,355/452,8479/8576,4702/4703,1,1,1,1,1,1,1]

[0,0.785398,0.988689,0.999787,1,1,1,1,1,1,1]
```

i.e. we get the right value ($\tan(\frac{\pi}{4})=1$) since f_5.

2) Results of the evaluation of the BF(z) (See Appendix D).

Appendix D

We have Appendix B loaded and we proceed in order to evaluate the Spherical Bessel functions of the second kind of all orders below Nmax = 5 using the proposed algorithm

```
#16:  YNEW(5, 2)
```

approximate #16

```
#17:  [0.2080734182, -0.3506120042, -0.7339914246, -1.484366557, -4.461291526, -18.59144531]
```

In 1995 an accuracy of 6 digits was used:

```
#18:  [0.208073, -0.350612, -0.733991, -1.48436, -4.46129, -18.5914]
```

We evaluate the Sph. B.f. of the 2nd kind of all orders below Nmax using
the utility file BesselFunctions.mth (approximated).

#19: YOLD(5, z)

#20: $$\left[-\frac{\cos(z)}{z}, -\frac{\cos(z)}{z^2}-\frac{\sin(z)}{z}, \frac{(z^2-3)\cdot\cos(z)}{z^3}-\frac{3\cdot\sin(z)}{z^2}, \frac{3\cdot(2\cdot z^2-5)\cdot\cos(z)}{z^4}+\frac{(z^2-15)\cdot\sin(z)}{z^3}, \frac{5\cdot(2\cdot z^2-21)\cdot\sin(z)}{z^4}-\frac{(z^4-45\cdot z^2+105)\cdot\cos(z)}{z^5}, -\frac{15\cdot(z^4-28\cdot z^2+63)\cdot\cos(z)}{z^6}-\frac{(z^4-105\cdot z^2+945)\cdot\sin(z)}{z^5}\right]$$

We substitute for z = 2 and approximate

#21: $$\left[-\frac{\cos(2)}{2}, -\frac{\cos(2)}{4}-\frac{\sin(2)}{2}, \frac{\cos(2)}{8}-\frac{3\cdot\sin(2)}{4}, \frac{9\cdot\cos(2)}{16}-\frac{11\cdot\sin(2)}{8}, \frac{59\cdot\cos(2)}{32}-\frac{65\cdot\sin(2)}{16}, \frac{495\cdot\cos(2)}{64}-\frac{541\cdot\sin(2)}{32}\right]$$

#22: [0.2080734182, -0.3506120042, -0.7339914246, -1.484366557, -4.461291526, -18.59144531]

with 6 digits accuracy as it was in 1995:

#23: [0.208073, -0.350612, -0.733991, -1.48436, -4.46129, -18.5914]

We evaluate up to seven terms in the continued fractions expression of
H(z=2) with Nmax = 5 (first approximated).

#24: H(7, 5, 2)

#25: [0, 0.1818181818, 0.1870503597, 0.1871631553, 0.1871649922, 0.1871650156, 0.1871650158, 0.1871650158]

with 6 digits accuracy as it was in 1995:

#26: [0, 0.181818, 0.18705, 0.187163, 0.187164, 0.187165, 0.187165, 0.187165]

We define nlim.

#27: nlim := 5

We evaluate the Sph. B.f. of the first kind of order Nmax-1 (approx).

#28: JTOPM1(5, 2)

#29: 0.01407939267

We evaluate the Sph. B.f. of the first kind of order Nmax (approx).

#30: JTOP(5, 2)

#31: 0.002635169421

We evaluate the Sph. B.f. of the 1st kind of all orders below Nmax using
the proposed algorithm (approximated).

#32: JNEW(5, 2)

#33: [0.002635169421, 0.01407939267, 0.06072209763, 0.198447949, 0.4353977749, 0.4546487134]

Accuracy 6 digits as it was in the original 1995 contribution

#34: [0.00263516, 0.0140793, 0.0607221, 0.198447, 0.435397, 0.454648]

We evaluate the Sph. B.f. of the 1st kind of all orders below Nmax using
the utility file BesselFunctions.mth, substitute for z = 2 and finally approximate.

#35: JOLD(5, z)

#36:
$$\left[\frac{15\cdot(z^4-28\cdot z^2+63)\cdot\text{SIN}(z)}{z^6}-\frac{(z^4-105\cdot z^2+945)\cdot\text{COS}(z)}{z^5},\ \frac{5\cdot(2\cdot z^2-21)\cdot\text{COS}(z)}{z^4}+\frac{(z^4-45\cdot z^2+105)\cdot\text{SIN}(z)}{z^5},\ \frac{(z^2-15)\cdot\text{COS}(z)}{z^3}+\frac{3\cdot(5-2\cdot z^2)\cdot\text{SIN}(z)}{z^4},\ \frac{(3-z^2)\cdot\text{SIN}(z)}{z^3}-\frac{3\cdot\text{COS}(z)}{z^2},\ \frac{\text{SIN}(z)}{z^2}-\frac{\text{COS}(z)}{z},\ \frac{\text{SIN}(z)}{z}\right]$$

#37:
$$\left[-\frac{541\cdot\text{COS}(2)}{32}-\frac{495\cdot\text{SIN}(2)}{64},\ -\frac{65\cdot\text{COS}(2)}{16}-\frac{59\cdot\text{SIN}(2)}{32},\ -\frac{11\cdot\text{COS}(2)}{8}-\frac{9\cdot\text{SIN}(2)}{16},\ -\frac{3\cdot\text{COS}(2)}{4}-\frac{\text{SIN}(2)}{8},\ \frac{\text{SIN}(2)}{4}-\frac{\text{COS}(2)}{2},\ \frac{\text{SIN}(2)}{2}\right]$$

#38: [0.002635169721, 0.01407939275, 0.06072209765, 0.198447949, 0.4353977749, 0.4546487134]

In 6 digits accuracy as it was in 1995:

#39: [0.0026324, 0.0140787, 0.0607218, 0.198447, 0.435397, 0.454648]

In the following table we present some results:

Function	Our Algorithm	BESSEL.MTH(95)	Reference [5]	BesselFunctions.mth (08)
$j_5(z=2)$	0.002635169421	0.0026324	0.0026352	0.002635169721
$j_4(z=2)$	0.01407939267	0.0140787	0.014079	0.01407939275
$j_3(z=2)$	0.06072209763	0.0607218	0.060722	0.06072209765
$j_2(z=2)$	0.198447949	0.198447	0.198447	0.198447949
$j_1(z=2)$	0.4353977749	0.435397	0.435397	0.4353977749
$j_0(z=2)$	0.4546487134	0.454648	0.454648	0.4546487134

These results are obtained with the same Derive precision (the Derive precision by default). *(This was 6 digits in 1995 and is 10 digits now with Derive 6.)* We observe that our algorithm produces better results than the ones obtained with the BESSEL.MTH file although the proposed algorithm is slower.

3) Results of the evaluation of the B(z) (See Appendix E).

The standard BESSEL_Y(k_,z) function cannot give results for the case $k\_ = 0$ and $z = 2$, but we get accurate results with our procedure [5] as it is shown in Appendix E.

Appendix E

We have Appendix C loaded and we proceed in order to evaluate the
Bessel function of the second kind of order 0 up to 10 terms in the summation.
We take z = 2. (all results are approximated!)

#20: VECTOR_Y0(10, 2)

#21: [0.6366197723, 0.489754084, 0.5119671213, 0.5103024958, 0.5103779228, 0.5103756229, 0.5103756734, 0.5103756726, 0.5103756726, 0.5103756726]

We define the Bessel function of the 2nd kind of order 0:

```
#22:  y0 := 0.5103756726

#23:  VECTOR_Y1(10, 2)

#24:  [-0.05499896203, -0.1127961327, -0.1066902521, -0.107045282, -0.1070320968,
       -0.1070324379, -0.1070324314, -0.1070324315, -0.1070324315, -0.1070324315]
```

We define the Bessel function of the 2nd kind of order 1:

```
#25:  y1 := -0.1070324315
```

We evaluate up to 10 terms in the continued fractions expression of H(z=2) with Nmax = 5:

```
#26:  H(10, 5, 2)

#27:  [0, 0.2, 0.2068965517, 0.207070707, 0.2070739549, 0.2070740015, 0.2070740021,
       0.2070740021, 0.2070740021, 0.2070740021, 0.2070740021]
```

We define nlim:

```
#28:  nlim := 6
```

We evaluate the Bessel function of the 1st kind of order Nmax:

```
#29:  JTOP(5, 2) = 0.007039629737
```

We evaluate the Bessel function of the 1st kind of order Nmax-1:

```
#30:  JTOPM1(5, 2) = 0.0339957198
```

We evaluate the Bessel functions of the 1st kind of all orders
below Nmax using the proposed algorithm

```
#31:  JNEW(5, 2)

#32:  [0.007039629737, 0.0339957198, 0.1289432494, 0.3528340286, 0.5767248078,
       0.2238907791]
```

We evaluate the Bessel functions of the 1st kind of all orders
below Nmax using the BesselFunctions.mth utility file.

```
#33:  JOLD(5, 2)

#34:  [0.007039629755, 0.0339957198, 0.1289432494, 0.3528340286, 0.5767248077,
       0.2238907791]
```

7 Conclusions

We have introduced the Derive version of an algorithm to evaluate the continued fraction representation of a rational approximation of a function and an efficient algorithm to evaluate BFs. This code illustrates the use of the continued fraction method and gives an accurate way to evaluate the BFs although it is slower than the implemented functions in BESSEL.MTH. The proposed method not only is very accurate but solves problems as the calculus of, for example, $Y_0(z = 2)$, that is not possible to obtain the `BESSEL_Y(0,2)` function. Moreover, the aim of this paper is to introduce in an interactive way a complete tested algorithm that can be presented as a tutorial in Bessel functions, claiming in the stability of the used recurrence relation (i.e. forward recurrence relations for the BFs of second kind and backward ones for the BFs of the first kind) and about the properties of convergence of the continued fractions approximations.

Comment in 2008: Times have changed since 2008. The complaint about `BESSEL_Y(0,2)` *is obsolete as you can see below. BesselFunctions.mth might be an improved version of BESSEL.MTH or it might have been that the authors would have achieved better results with the implemented functions by increasing the precision. Josef.*

```
BESSEL_Y(0, 2)

0.5103756726
```

References

[1] Yu. L. Ratis, P. Fernández de Córdoba, *A code to calculate (high order) Bessel functions based on the continued fraction method.* Comput. Phys. Commun. 76 (1993) 381-388.

[2] Y.L. Luke, *Mathematical Functions and their Approximations*. Academic, New York, (1975).

[3] J. C. P. Miller, *Bessel Functions, Part II, Functions of Positive Integer Order, Mathematical Tables, Vol. 10.* Cambridge University Press, (1952).

[4] W. H. Press, B. P. Flannery, S. A. Teukolsky, W. T. Vetterling, *Numerical Recipes.* Cambridge University Press (1989).

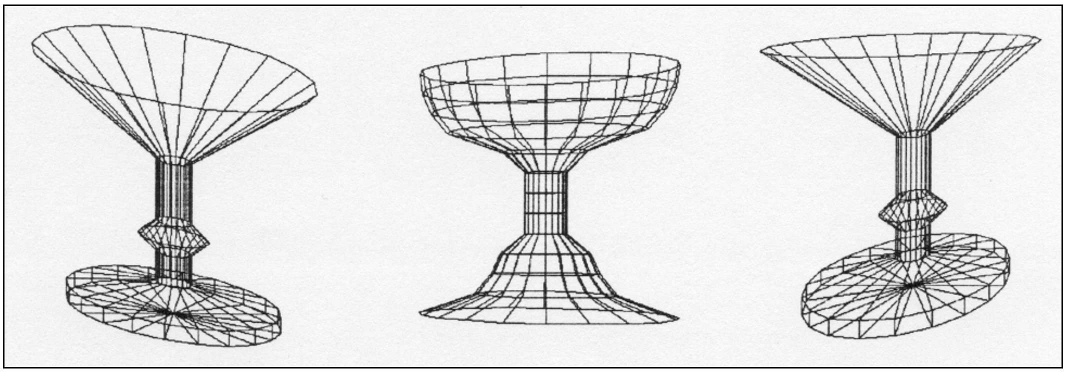
[5] M. Abramowitz, I. A. Stegun, Eds. *Handbook of Mathematical Functions*. Dover (1972).

Additional comment:

If you want to know more about Bessel functions and their applications, you can find a lot of information in the web. Bessel functions are therefore especially important for many problems of wave propagation, static potentials, and so on. For example: electromagnetic waves, heat conduction, vibration of membranes, diffusion problems, signal processing. Crick and Watson used BFs to develop their model of the DNA double helix.

Three pictures from my Graphic-Workshop held at the DERIVE Days Düsseldorf (DERIVE for DOS).

Surfaces of Revolution in Parallel- and Central Projection



Cubic Splines

Josef Böhm, Otto Reichel, Leo Klingen

There are a lot of problems dealing with the task to find a curve connecting more or less given points – nodes – in the plane (e.g. road construction, design forms, …). Let us formulate the interpolation problem: which continuous and differentiable – not too complex – curve connects n given nodes (x_k, y_k) with $k = 1, \ldots, n$.

Example:

k	x_k	y_k	$x_{k+1}-x_k = h_k$	$y_{k+1}-y_k = j_k$
1	0	1	1	1
2	1	2	2	2
3	3	4	1	–1.5
4	4	2.5	1	–2.5
5	5	0	1	1
6	6	1		

The easiest way to find a fitting function would be to find a polynomial of order n–1:

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots + a_1 x + a_0. \qquad (*)$$

The unknown coefficients $a_0, a_1, \ldots, a_{n-1}$ are the solution of a system of n linear equations:

$$[p(x_1) = y_1, p(x_2) = y_2, \ldots, p(x_n) = y_n].$$

Doing this in case of more nodes we might face some disadvantages:

- a huge system of equation must be solved (could be done using a powerful CAS),
- it is not very comfortable to work with a polynomial of high order (no problem with a CAS),
- the curve would show up to n–2 turning points, i.e. the curve could be very oscillating (this can not be changed using a CAS). We prefer a smoother curve.

So let´s try another approach tackling the problem. We connect neighbouring points by cubics

$$p_k(x) = a_k + b_k(x - x_k) + c_k(x - x_k)^2 + d_k(x - x_k)^3$$

in such a way that they have a node in common (same function values), have the same first derivative in this node (then curve is differentiable in all points) and have also the second derivative in common (which is responsible that there is no "shock" in the curvature).

(This is a welcome occasion to demonstrate the students the advantage of developing a polynomial on a certain position.)

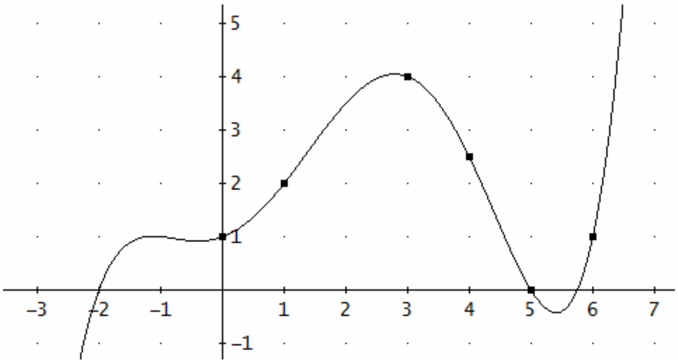
The easiest way to find the polynomial (∗) is applying the FIT-function:

Let´s do this as an intro and then proceed with the cubic splines.

#1: nodes := $\begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \\ 4 & 2.5 \\ 5 & 0 \\ 6 & 1 \end{bmatrix}$

#2: FIT($\left[x,\ a\cdot x^5 + b\cdot x^4 + c\cdot x^3 + d\cdot x^2 + e\cdot x + f\right]$, nodes)

#3: $0.0125\cdot x^5 - 0.1041666666\cdot x^4 + 0.0625\cdot x^3 + 0.6041666666\cdot x^2 + 0.425\cdot x + 1$



We set up 4 linear equations for each of the $n - 1$ cubics ((1) – (4)). We lack two equations to get a unique solution for the system. So we add two meaningful equations. Setting the 2nd derivative equal zero in start and end point is one popular possibility (*natural splines*) (equations (5) and (6)). Other settings are possible.

System I:

$p_k(x_k) = y_k$	$k = 1,\ldots,n-1$	(1)	
$p_k(x_{k+1}) = y_{k+1}$	$k = 1,\ldots,n-1$	(2)	
$p_k'(x_{k+1}) = p_{k+1}'(x_{k+1})$	$k = 1,\ldots,n-2$	(3)	!!!
$p_k''(x_{k+1}) = p_{k+1}''(x_{k+1})$	$k = 1,\ldots,n-2$	(4)	!!!
$p_1''(x_1) = 0$		(5)	
$p_{n-1}''(x_n) = 0$		(6)	

and with

$$p_k'(x) = b_k + 2c_k(x - x_k) + 3d_k(x - x_k)^2$$

$$p_k''(x) = 2c_k + 6d_k(x - x_k) \quad \text{and} \quad h_k = x_{k+1} - x_k$$

we obtain

System II:

(1)	$a_k = y_k$	$k = 1,\ldots,n-1$
(2)	$a_k + b_k h_k + c_k h_k^2 + d_k h_k^3 = y_{k+1}$	$k = 1,\ldots,n-1$
(3)	$b_k + 2c_k h_k + 3d_k h_k^2 = b_{k+1}$	$k = 1,\ldots,n-2$
(4)	$2c_k + 6d_k h_k = 2c_{k+1}$	$k = 1,\ldots,n-2$
(5)	$c_1 = 0$	
(6)	$2c_{n-1} + 6d_{n-1}h_{n-1} = 0$	

For our special problem using the data points System II leads us immediately to **System III**:

(1) $a_1 = 1,\ a_2 = 2,\ a_3 = 4,\ a_4 = 2.5,\ a_5 = 0$

(2) $a_1 + b_1 + c_1 + d_1 = 2$

$a_2 + 2b_2 + 4c_2 + 8d_2 = 4$

$a_3 + b_3 + c_3 + d_3 = 2.5$

$a_4 + b_4 + c_4 + d_4 = 0$

$a_5 + b_5 + c_5 + d_5 = 1$

(3) $b_1 + 2c_1 + 3d_1 = b_2$

$b_2 + 4c_2 + 12d_2 = b_3$

$b_3 + 2c_3 + 3d_3 = b_4$

$b_4 + 2c_4 + 3d_4 = b_5$

(4) $2c_1 + 6d_1 = 2c_2$

$2c_2 + 12d_2 = 2c_3$

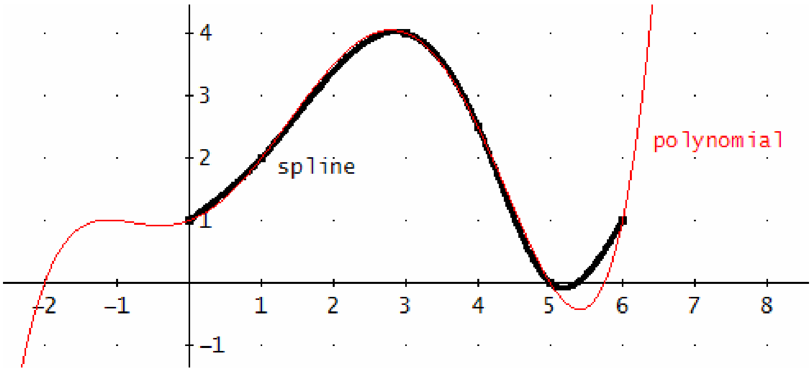
$2c_3 + 6d_3 = 2c_4$

$2c_4 + 6d_4 = 2c_5$

(5) $c_1 = 0$

(6) $2c_5 + 6d_5 = 0$

We have 4·(6 – 1) = 20 equations. There are $4n – 4$ coefficients but only $3n – 4$ are really unknown. This system can easily be solved – not by hand but by using DERIVE. We obtain the coefficients of the $p_k(x)$ functions valid for the intervals $x_k \le x \le x_{k+1}$. Either doing it by hand or using a DERIVE function we create the cubics.



SPLI_1_new.mth

```
#1:   InputMode := Word

#2:   [nodes := [], coeff :=]

      xk := VECTOR(nodes    , k, DIM(nodes))
#3:                     k,1

               ⎛ 3                                               i                  ⎞
#4:   pk := VECTOR⎜ ∑  coeff                          ·(x - xk   ) , k_, DIM(xk) - 1⎟
               ⎝i=0       (DIM(xk) - 1)·i + k_             k_                      ⎠

                 DIM(xk) - 1
#5:   SPL(x) :=      ∑        pk ·χ(xk , x, xk     )
                    j=1         j     j      j + 1

      TAB(incr) := VECTOR([x, SPL(x)], x, nodes    , nodes             , incr)
#6:                                            1,1        DIM(nodes),1
```

#7: Instead of TAB we use now the TABLE-function (see #23 and #26).

#8: Enter the matrix of the given points and name the matrix nodes.

#9: coeff is the solution vector – see below.

#10: pk returns the separated cubic splines and

#11: SPL(x) gives the complete interpolating function.

#12: nodes :=
$$\begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \\ 4 & 2.5 \\ 5 & 0 \\ 6 & 1 \end{bmatrix}$$

#13: The system:

#14:
$$[a1 = 1,\ a2 = 2,\ a3 = 4,\ a4 = 2.5,\ a5 = 0,\ c1 = 0,\ a1 + b1 + c1 + d1 = 2,\ a2 + 2\cdot b2 + 4\cdot c2 + 8\cdot d2 = 4,\ a3 + b3 + c3 + d3 = 2.5,\ a4 + b4 + c4 + d4 = 0,\ a5 + b5 + c5 + d5 = 1,\ b1 + 2\cdot c1 + 3\cdot d1 = b2,\ b2 + 4\cdot c2 + 12\cdot d2 = b3,\ b3 + 2\cdot c3 + 3\cdot d3 = b4,\ b4 + 2\cdot c4 + 3\cdot d4 = b5,\ 2\cdot c1 + 6\cdot d1 = 2\cdot c2,\ 2\cdot c2 + 12\cdot d2 = 2\cdot c3,\ 2\cdot c3 + 6\cdot d3 = 2\cdot c4,\ 2\cdot c4 + 6\cdot d4 = 2\cdot c5,\ 2\cdot c5 + 6\cdot d5 = 0]$$

#15:
$$coeff := (\mathrm{SOLUTIONS}([a1 = 1,\ a2 = 2,\ a3 = 4,\ a4 = 2.5,\ a5 = 0,\ c1 = 0,\ a1 + b1 + c1 + d1 = 2,\ a2 + 2\cdot b2 + 4\cdot c2 + 8\cdot d2 = 4,\ a3 + b3 + c3 + d3 = 2.5,\ a4 + b4 + c4 + d4 = 0,\ a5 + b5 + c5 + d5 = 1,\ b1 + 2\cdot c1 + 3\cdot d1 = b2,\ b2 + 4\cdot c2 + 12\cdot d2 = b3,\ b3 + 2\cdot c3 + 3\cdot d3 = b4,\ b4 + 2\cdot c4 + 3\cdot d4 = b5,\ 2\cdot c1 + 6\cdot d1 = 2\cdot c2,\ 2\cdot c2 + 12\cdot d2 = 2\cdot c3,\ 2\cdot c3 + 6\cdot d3 = 2\cdot c4,\ 2\cdot c4 + 6\cdot d4 = 2\cdot c5,\ 2\cdot c5 + 6\cdot d5 = 0],\ [a1,\ a2,\ a3,\ a4,\ a5,\ b1,\ b2,\ b3,\ b4,\ b5,\ c1,\ c2,\ c3,\ c4,\ c5,\ d1,\ d2,\ d3,\ d4,\ d5]))_1$$

#16:
$$coeff := \left[1,\ 2,\ 4,\ \frac{5}{2},\ 0,\ \frac{33}{38},\ \frac{24}{19},\ -\frac{6}{19},\ -\frac{51}{19},\ -\frac{18}{19},\ 0,\ \frac{15}{38},\ -\frac{45}{38},\ -\frac{45}{38},\ \frac{111}{38},\ \frac{5}{38},\ -\frac{5}{19},\ 0,\ \frac{26}{19},\ -\frac{37}{38}\right]$$

#17: pk

#18:
$$\left[\frac{5\cdot x^3}{38} + \frac{33\cdot x}{38} + 1,\ -\frac{10\cdot x^3 - 45\cdot x^2 + 12\cdot x - 53}{38},\ -\frac{45\cdot x^2 - 258\cdot x + 217}{38},\ \frac{52\cdot x^3 - 669\cdot x^2 + 2754\cdot x - 3545}{38},\ \frac{(5 - x)\cdot(37\cdot x^2 - 481\cdot x + 1516)}{38}\right]$$

#19: SPL(x)

#20: $$\frac{(x-5)\cdot(37\cdot x^2-481\cdot x+1516)\cdot \mathrm{SIGN}(x-6)}{76}-\frac{89\cdot(x^2-10\cdot x+25)\cdot|x-5|}{76}+\frac{13\cdot(x^2-8\cdot x+16)\cdot|x-4|}{19}+\frac{5\cdot(x^2-6\cdot x+9)\cdot|x-3|}{38}-\frac{15\cdot(x^2-2\cdot x+1)\cdot|x-1|}{76}+\frac{(5\cdot x^3+33\cdot x+38)\cdot \mathrm{SIGN}(x)}{76}$$

#21: TAB(0.5)

#22:

0	±0.5 + 0.5
0.5	1.450657894
1	?
1.5	2.697368421
2	3.394736842
2.5	3.894736842
3	?
3.5	3.546052631
4	?
4.5	1.032894736
5	0
5.5	0.1348684210
6	±0.5 + 0.5

#23: TABLE(SPL(x), x, 0, 6, 0.5)

#24:

0	±0.5 + 0.5
0.5	1.450657894
1	?
1.5	2.697368421
2	3.394736842
2.5	3.894736842
3	?
3.5	3.546052631
4	?
4.5	1.032894736
5	0
5.5	0.1348684210
6	±0.5 + 0.5

The function values in the nodes are strange – this is a consequence of using the CHI-function. We could overcome this problem by using an IF-construct to define the piecewise defined function.

It is interesting that we can close the gaps by redefining the function:

#25: $$\mathrm{SPL\_}(x) := \frac{(x-5)\cdot(37\cdot x^2-481\cdot x+1516)\cdot \mathrm{SIGN}(x-6)}{76}-\frac{89\cdot(x^2-10\cdot x+25)\cdot|x-5|}{76}+\frac{13\cdot(x^2-8\cdot x+16)\cdot|x-4|}{19}+\frac{5\cdot(x^2-6\cdot x+9)\cdot|x-3|}{38}-\frac{15\cdot(x^2-2\cdot x+1)\cdot|x-1|}{76}+\frac{(5\cdot x^3+33\cdot x+38)\cdot \mathrm{SIGN}(x)}{76}$$

```
#26:  TABLE(SPL_(x), x, 0, 6, 0.5)
```

#27:

0	±0.5 + 0.5
0.5	1.450657894
1	2
1.5	2.697368421
2	3.394736842
2.5	3.894736842
3	4
3.5	3.546052631
4	2.5
4.5	1.032894736
5	0
5.5	0.1348684210
6	±0.5 + 0.5

```
#28:  The polynomial of order 5 through the nodes:
```

#29: $\text{FIT}(\left[x,\ a\cdot x^5 + b\cdot x^4 + c\cdot x^3 + d\cdot x^2 + e\cdot x + f\right],\ \text{nodes})$

#30: $$\frac{x^5}{80} - \frac{5\cdot x^4}{48} + \frac{x^3}{16} + \frac{29\cdot x^2}{48} + \frac{17\cdot x}{40} + 1$$

```
#31:  You can load this file as a utility file!

#32:  TABLE(SPL_(x), x, 0, 6, 0.01)
```

The last expressions is the base for plotting the spline as a thick line (Points connected and size large).

But you might prefer another way. The next procedure is creating the system of equations, too: `SPLI_2_new.mth` and `SPLI_2_new.dfw`

#1: $d(a) := \text{DIM}(a) - 1$

#2: $\left[x\_(a,\ k) := a_{k,1},\ y\_(a,\ k) := a_{k,2}\right]$

#3: $h(a,\ k) := x\_(a,\ k + 1) - x\_(a,\ k)$

```
#4:  m0(a) := VECTOR(VECTOR(IF(k = 4·j - 3, 1, 0), k, 4·d(a)), j, d(a))
```

#5: $\text{m1}(a) := \text{VECTOR}(\text{VECTOR}(\text{IF}(k = 4\cdot j - 3,\ 1,\ \text{IF}(k = 4\cdot j - 2,\ h(a,\ j),\ \text{IF}(k = 4\cdot j - 1,\ h(a,\ j)^2,\ \text{IF}(k = 4\cdot j,\ h(a,\ j)^3,\ 0)))),\ k,\ 4\cdot d(a)),\ j,\ d(a))$

#6: $\text{m2}(a) := \text{VECTOR}(\text{VECTOR}(\text{IF}(k = 4\cdot j - 2,\ 1,\ \text{IF}(k = 4\cdot j - 1,\ 2\cdot h(a,\ j),\ \text{IF}(k = 4\cdot j,\ 3\cdot h(a,\ j)^2,\ \text{IF}(k = 4\cdot j + 2,\ -1,\ 0)))),\ k,\ 4\cdot d(a)),\ j,\ d(a) - 1)$

$$\#7: \quad \mathrm{m3(a)} := \mathrm{VECTOR(VECTOR(IF}(k = 4\cdot j - 1,\ 2,\ \mathrm{IF}(k = 4\cdot j,\ 6\cdot \mathrm{h}(a, j),\ \mathrm{IF}(k = 4\cdot j + 3,\ -2,\ 0))),\ k,\ 4\cdot \mathrm{d}(a)),\ j,\ \mathrm{d}(a))$$

$$\#8: \quad \mathrm{m4(a)} := [\mathrm{VECTOR(IF}(j = 3,\ 1,\ 0),\ j,\ 4\cdot \mathrm{d}(a))]$$

$$\#9: \quad \mathrm{system(a)} := \mathrm{APPEND(m0(a),\ m1(a),\ m2(a),\ m3(a),\ m4(a))}$$

$$\#10: \quad \mathrm{c0(a)} := \mathrm{VECTOR(y\_(a,\ k),\ k,\ d(a))}$$

$$\#11: \quad \mathrm{c1(a)} := \mathrm{VECTOR(y\_(a,\ k + 1),\ k,\ d(a))}$$

$$\#12: \quad \mathrm{c2(a)} := \mathrm{VECTOR(0,\ k,\ 2\cdot d(a))}$$

$$\#13: \quad \mathrm{c(a)} := \mathrm{APPEND(c0(a),\ c1(a),\ c2(a))}$$

$$\#14: \quad \mathrm{coeff(a)} := \left[\mathrm{system(a)}^{-1}\cdot \mathrm{c(a)}\right]'$$

$$\#15: \quad \mathrm{k\_(a,\ k)} := (\mathrm{coeff(a)})_k$$

$$\#16: \quad \mathrm{spl(a)} := \mathrm{VECTOR}\left(\sum_{i=0}^{3} \mathrm{k\_(a,\ 4\cdot j - i)}\cdot(x - \mathrm{x\_(a,\ j)})^{3 - i},\ j,\ \mathrm{d(a)}\right)$$

$$\#17: \quad \mathrm{splines(a)} := \sum_{i=1}^{\mathrm{d(a)}} (\mathrm{spl(a)})_{i,1}\cdot \chi(\mathrm{x\_(a,\ i)},\ x,\ \mathrm{x\_(a,\ i + 1)})$$

$$\#18: \quad \left[\mathrm{nodes} := \begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \\ 4 & 2.5 \\ 5 & 0 \\ 6 & 1 \end{bmatrix},\ \mathrm{pts} := \begin{bmatrix} -6 & 1 \\ -4 & -3.5 \\ -2 & -3 \\ 1 & 5.5 \\ 2 & 7 \\ 3.5 & 2 \\ 5 & 2 \\ 8 & 0 \\ 10 & -1.5 \end{bmatrix}\right]$$

We have two examples: point matrix `nodes` and point matrix `pts`.

$$\#19: \quad \mathrm{splines(nodes)}$$

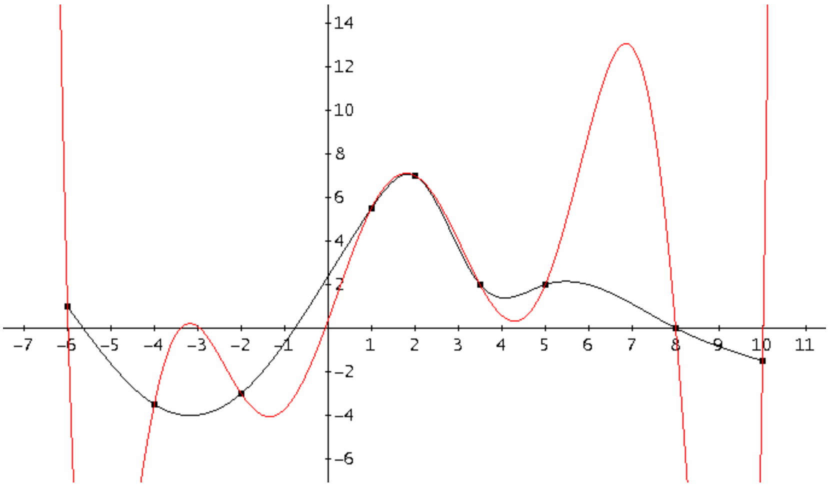
$$\#20: \quad \frac{(x - 5)\cdot(37\cdot x^2 - 481\cdot x + 1516)\cdot \mathrm{SIGN}(x - 6)}{76} - \frac{89\cdot(x^3 - 15\cdot x^2 + 75\cdot x - 125)\cdot \mathrm{SIGN}(x - 5)}{76} + \frac{13\cdot(x^3 - 12\cdot x^2 + 48\cdot x - 64)\cdot \mathrm{SIGN}(x - 4)}{19} + \frac{5\cdot(x^3 - 9\cdot x^2 + 27\cdot x - 27)\cdot \mathrm{SIGN}(x - 3)}{38} - \frac{15\cdot(x^3 - 3\cdot x^2 + 3\cdot x - 1)\cdot \mathrm{SIGN}(x - 1)}{76} + \frac{(5\cdot x^3 + 33\cdot x + 38)\cdot \mathrm{SIGN}(x)}{76}$$

$$\#21: \quad \mathrm{splines(pts)}$$

#20 gives the same result as before. I simplify #21 and plot the composed spline together with the interpolation polynomial.

$$\#23:\quad \text{FIT}\left(\left[x,\ a\cdot x^8 + b\cdot x^7 + c\_\cdot x^6 + d\_\cdot x^5 + e\cdot x^4 + f\cdot x^3 + g\cdot x^2 + h\_\cdot x + i\right],\ \text{pts}\right)$$

$$\#24:\quad 6.109773165\cdot 10^{-5}\cdot x^8 - 0.0008684255363\cdot x^7 - 0.001684734074\cdot x^6 + 0.05405121871\cdot x^5 - 0.03029723448\cdot x^4 - 0.9550514929\cdot x^3 + 0.5513712830\cdot x^2 + 5.510965706\cdot x + 0.3714525808$$



You might guess which one of the two curves is the spline and which one is the polynomial?

a represents the matrix of the points, **system(a)** gives the left hand side matrix of System III from above, **c0(a)** gives the right hand side of System III (the constants), **coeff(a)** returns the solutions, **spl(a)** shows the cubics in a vector and finally **splines(a)** creates the composed function valid for the whole interval.

With a bit of manipulation skills **System III** can be transformed into a special form. This could be done with the students and presents a fine example that even in times of a CAS some manipulation skills are very useful.

We would like to eliminate all the variables in **System II** except the c's. In the first two steps we eliminate a_k and d_k using (1) and (4) (page 28), then we set $y_{k+1} - y_k = j_k$. Now the two equations (2) and (3) are remaining (containing b_k and d_k):

(2) $\quad b_k h_k + c_k h_k^2 + \frac{c_{k+1} - c_k}{3} h_k^2 = j_k;\ \text{solve for } b_k \rightarrow b_k = \ldots \quad (*)$

(3) $\quad b_k + 2c_k h_k + (c_{k+1} - c_k) h_k = b_{k+1};\ \text{solve for } b_k \rightarrow b_k = \ldots$

We compare the two expressions for the b_k and solve the resulting equation for b_{k+1}:

$$b_{k+1} = \frac{j_k}{h_k} + c_{k+1}\, h_k - \frac{c_{k+1} - c_k}{3} h_k .$$

Decreasing the index leads to a new expression for b_k.

$$b_k = \frac{j_{k-1}}{h_{k-1}} + c_k\, h_{k-1} - \frac{c_k - c_{k-1}}{3} h_{k-1}.$$

If we now equate this expression with the expression (∗) for b_k we obtain **System IV**:

$$c_{k-1}\, h_{k-1} + 2c_k\,(h_k + h_{k-1}) + c_{k+1} h_k = 3\left(\frac{j_k}{h_k} - \frac{j_{k-1}}{h_{k-1}}\right);\ k = 2,\ldots,n-1$$

$$c_1 = c_n = 0$$

And in our case **System V** at last is hiding the spline problem´s solution:

$$\begin{array}{llll} 6c_2 + 2c_3 & = 0 & (k=2) \\ 2c_2 + 6c_3 + c_4 & = -7.5 & (k=3) \\ c_3 + 4c_4 + c_5 & = -3 & (k=4) \\ c_4 + 4c_5 & = 10.5 & (k=5) \end{array}$$

For solving this symmetric "tridiagonal" system exists a fast algorithm even for large systems and having found the c_k-coefficients it is not difficult to find the values for the b_k , the a_k *and* d_k. (See references).

For the real DERIVIANs it might be nice to perform the manipulations from above using the CAS. Students could check their competence in manually manipulations and DERIVE manipulations as well:
I substitute b_k = b_:

#1: $\text{SOLVE}\left(b\_\cdot h_k + c_k\cdot h_k^2 + \frac{c_{k+1} - c_k}{3}\cdot h_k^2 = j_k,\ b\_\right)$

#2: $b\_ = -\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k}$

#3: $\text{SOLVE}(b\_ + 2\cdot c_k\cdot h_k + (c_{k+1} - c_k)\cdot h_k = b_{k+1},\ b\_)$

#4: $b\_ = b_{k+1} - c_{k+1}\cdot h_k - c_k\cdot h_k$

#5: $-\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k} = b_{k+1} - c_{k+1}\cdot h_k - c_k\cdot h_k$

I substitute b_{k+1} = b__. (In Simplify Menu: Subexpression Substitution)

#6: $-\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k} = b\_\_ - c_{k+1}\cdot h_k - c_k\cdot h_k$

#7: $\text{SOLVE}\left(-\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k} = b\_\_ - c_{k+1}\cdot h_k - c_k\cdot h_k,\ b\_\_\right)$

#8: $b\_\_ = \frac{2\cdot c_{k+1}\cdot h_k^2 + c_k\cdot h_k^2 + 3\cdot j_k}{3\cdot h_k}$

I substitute b__ → b_ and k → k – 1 and then simplify expression #10:

#9: $b\_ = \frac{2\cdot c_{(k-1)+1}\cdot h_{k-1}^2 + c_{k-1}\cdot h_{k-1}^2 + 3\cdot j_{k-1}}{3\cdot h_{k-1}}$

#10: $$b\_ = \frac{c_{k-1}\cdot h_{k-1}^2 + 2\cdot c_k\cdot h_{k-1}^2 + 3\cdot j_{k-1}}{3\cdot h_{k-1}}$$

#11: $$-\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k} = \frac{c_{k-1}\cdot h_{k-1}^2 + 2\cdot c_k\cdot h_{k-1}^2 + 3\cdot j_{k-1}}{3\cdot h_{k-1}}$$

#12: $$\text{sysIV} := -\frac{2\cdot c_k\cdot h_k^2 + c_{k+1}\cdot h_k^2 - 3\cdot j_k}{3\cdot h_k} = \frac{c_{k-1}\cdot h_{k-1}^2 + 2\cdot c_k\cdot h_{k-1}^2 + 3\cdot j_{k-1}}{3\cdot h_{k-1}}$$

Expression #13 creates the system. h and j are vectors obtained from the table page 27.

```
#13:  VECTOR(sysIV, k, 2, 5)

#14:  [h := [1, 2, 1, 1, 1], j := [1, 2, -1.5, -2.5, 1]]
```

#15: $$\left[c_3 + 2\cdot c_2 = -\frac{2\cdot c_2 + c_1}{2},\ c_4 + 2\cdot c_3 = -\frac{8\cdot c_3 + 4\cdot c_2 + 15}{2},\ c_5 + 2\cdot c_4 = -2\cdot c_4 - c_3 - 3,\ c_6 + 2\cdot c_5 = -\frac{4\cdot c_5 + 2\cdot c_4 - 21}{2}\right]$$

I substitute $c_1 = c_6 = 0$:

#17: $$\left[c_3 + 2\cdot c_2 = -\frac{2\cdot c_2 + 0}{2},\ c_4 + 2\cdot c_3 = -\frac{8\cdot c_3 + 4\cdot c_2 + 15}{2},\ c_5 + 2\cdot c_4 = -2\cdot c_4 - c_3 - 3,\ 0 + 2\cdot c_5 = -\frac{4\cdot c_5 + 2\cdot c_4 - 21}{2}\right]$$

#18: $$\left[c_3 + 2\cdot c_2 = -c_2,\ c_4 + 2\cdot c_3 = -\frac{8\cdot c_3 + 4\cdot c_2 + 15}{2},\ c_5 + 2\cdot c_4 = -2\cdot c_4 - c_3 - 3,\ c_5 = -\frac{4\cdot c_5 + 2\cdot c_4 - 21}{4}\right]$$

Unfortunately we can not recognize the nice "tridiagonal" form in expression #18. But we try to solve this system for the c_i. I replace the c_i by c_i and solve the system:

```
#19:  InputMode := Word
```

#20: $$\text{SOLVE}\left(\left[c\_3 + 2\cdot c\_2 = -c\_2,\ c\_4 + 2\cdot c\_3 = -\frac{8\cdot c\_3 + 4\cdot c\_2 + 15}{2},\ c\_5 + 2\cdot c\_4 = -2\cdot c\_4 - c\_3 - 3,\ c\_5 = -\frac{4\cdot c\_5 + 2\cdot c\_4 - 21}{4}\right],\ [c\_2, c\_3, c\_4, c\_5]\right)$$

#21: $$\left[c\_2 = \frac{15}{38} \wedge c\_3 = -\frac{45}{38} \wedge c\_4 = -\frac{45}{38} \wedge c\_5 = \frac{111}{38}\right]$$

Comparing this with the solution of SYSTEM V or with the respective values in the coeff-vector in expression #16 page 30 we can be proud that we did a good job manually and "CASially" as well.

I continue with transferring the said algorithm into a DERIVE code.

#1: file spli_3_new.mth

#2: nodes :=

#3: $h(k) := nodes_{k+1,1} - nodes_{k,1}$

#4: $f(k) := 2\cdot(h(k+1) + h(k))$

#5: $m(k) := \dfrac{nodes_{k+1,2} - nodes_{k,2}}{h(k)}$

#6: $g(k) := 3\cdot(m(k+1) - m(k))$

#7:
```
a(n) :=
  If n = 1
     f(1)
     f(n) - h(n)^2/a(n - 1)
```

#8: $a0 := APPEND(VECTOR(nodes_{i,2}, i, DIM(nodes) - 1), [nodes_{DIM(nodes),2}])$

#9: a_val := VECTOR(a(i), i, DIM(nodes) - 2)

#10:
```
b(n) :=
  If n = 1
     g(1)
     g(n) - b(n - 1)·h(n)/a(n - 1)
```

#11: b_val := VECTOR(b(i), i, DIM(nodes) - 2)

#12:
```
c_aux(n) :=
  If n = 1
     b(DIM(nodes) - 2)/a(DIM(nodes) - 2)
     (b(DIM(nodes) - n - 1) - h(DIM(nodes) - n)·c_aux(n - 1))/a(DIM(nodes) - n - 1)
```

#13: a2 := APPEND([0], VECTOR(c_aux(j), j, DIM(nodes) - 2, 1, -1), [0])

#14: $a3 := VECTOR\left(\dfrac{a2_{j+1} - a2_j}{3\cdot h(j)}, j, DIM(nodes) - 1\right)$

#15: $a1 := VECTOR\left(\dfrac{a0_{j+1} - a0_j}{h(j)} - \dfrac{h(j)\cdot(a2_{j+1} + 2\cdot a2_j)}{3}, j, DIM(nodes) - 1\right)$

#16: $splines := VECTOR\left(a0_j + a1_j\cdot(x - nodes_{j,1}) + a2_j\cdot(x - nodes_{j,1})^2 + a3_j\cdot(x - nodes_{j,1})^3, j, DIM(nodes) - 1\right)$

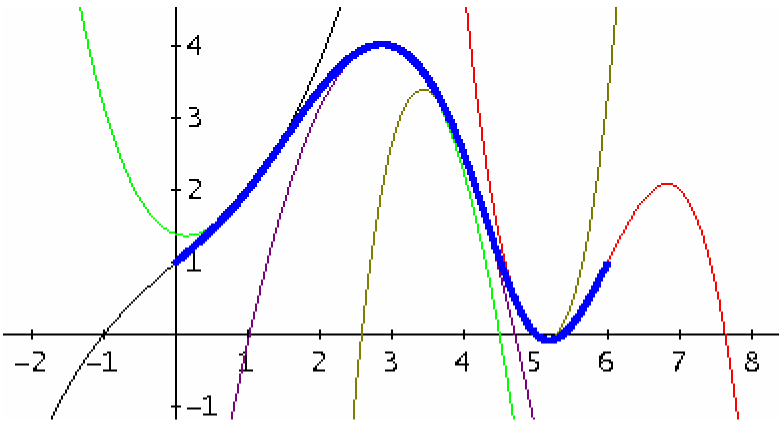
#17: $spl(x) := \sum_{i=1}^{DIM(nodes) - 1} splines_i\cdot\chi(nodes_{i,1}, x, nodes_{i+1,1})$

#18: $nodes := \begin{bmatrix} 0 & 1 \\ 1 & 2 \\ 3 & 4 \\ 4 & 2.5 \\ 5 & 0 \\ 6 & 1 \end{bmatrix}$

$$\#19:\quad splines = \left[\frac{5\cdot x^3}{38} + \frac{33\cdot x}{38} + 1,\ -\frac{10\cdot x^3 - 45\cdot x^2 + 12\cdot x - 53}{38},\ -\frac{45\cdot x^2 - 258\cdot x + 217}{38},\ \frac{52\cdot x^3 - 669\cdot x^2 + 2754\cdot x - 3545}{38},\ \frac{(5 - x)\cdot(37\cdot x^2 - 481\cdot x + 1516)}{38}\right]$$

$$\#20:\quad spl(x) = \frac{(x - 5)\cdot(37\cdot x^2 - 481\cdot x + 1516)\cdot SIGN(x - 6)}{76} - \frac{89\cdot(x^3 - 15\cdot x^2 + 75\cdot x - 125)\cdot SIGN(x - 5)}{76} + \frac{13\cdot(x^3 - 12\cdot x^2 + 48\cdot x - 64)\cdot SIGN(x - 4)}{19} + \frac{5\cdot(x^3 - 9\cdot x^2 + 27\cdot x - 27)\cdot SIGN(x - 3)}{38} - \frac{15\cdot(x^3 - 3\cdot x^2 + 3\cdot x - 1)\cdot SIGN(x - 1)}{76} + \frac{(5\cdot x^3 + 33\cdot x + 38)\cdot SIGN(x)}{76}$$

It is very informative to plot the single cubics in different colours – together with the resulting spline. One can see the very smooth transition from one cubic to the next one.



Starting with DERIVE 5 we could program and so we are able to collect the whole procedure into one program. As an extra you can find Don Phillip´s program among the accompanying files. Another program can be found in my book "Programmieren in Derive" (bk-teachware).

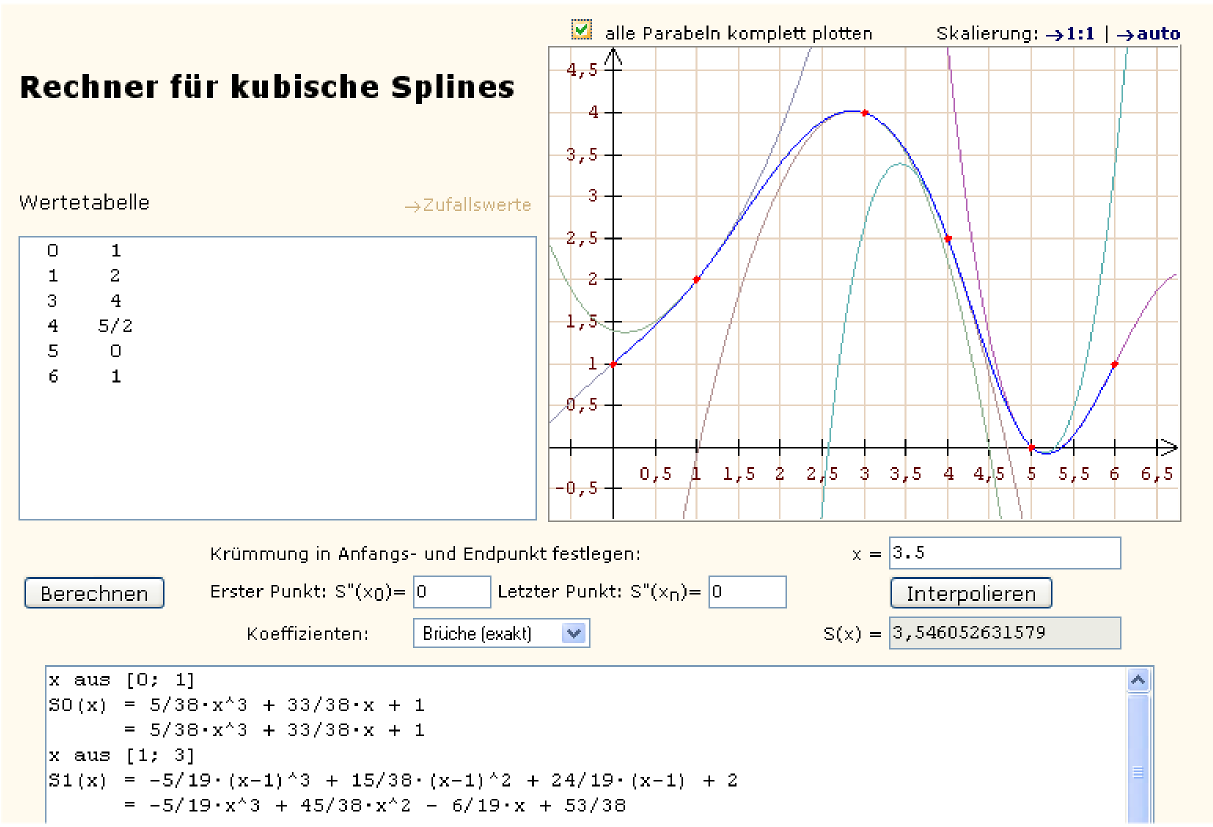
At last I´d like to add that I like the "cubic splines", because the "Reverse Curve Discussion" (in Austria: Umgekehrte Kurvendiskussion, in Germany "Steckbriefaufgaben" = "wanted circular problems") are a fixpoint in our school mathematics and I for my person have some problems answering questions about the importance of these problems in real life. I can assure you that students also are enthusiastic with the splines.

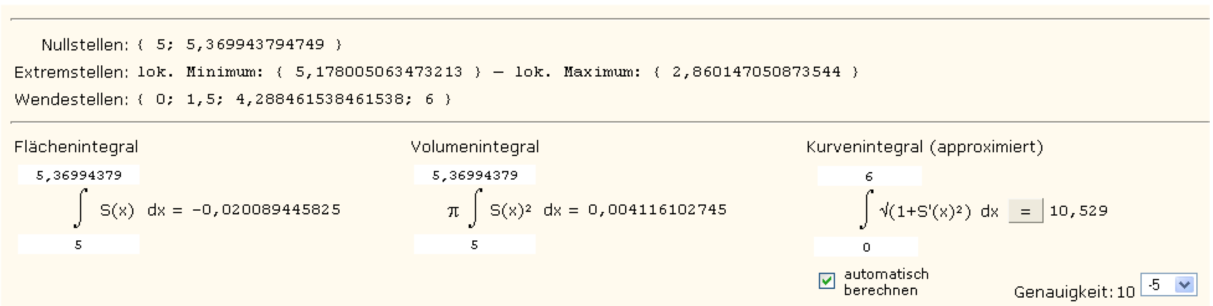
In DNL#19 I´ll show how Otto Reichel from St. Pölten and Leo Klingen from Bonn are dealing with this issue. As a special gift you will find Mr Klingen´s function to produce paramtetric splines. (Do you remember the Easter Bunny from last year?)

At the DERIVE Days Düsseldorf Günter Scheu promised to send a paper on Bézier Curves. And he really did. Thank you, Günter, you are very reliable. So we´ll see how this interpolation technique is working soon.

I found a nice website with an applet for calculating and presentating cubic splines:

http://www.arndt-bruenner.de/mathe/scripts/kubspline.htm





References

W. Luther & M. Ohsmann, *Mathematische Grundlagen der Computergraphik*, Vieweg, 1989

Josef Böhm, *Programmieren in Derive*, bk-teachware, 2002

Max-Günter Schröfel, *Cubic Spline Functions, Matrices and Derive*, Proceedings of DES-TIME Dresden, bk-teachware, 2006

http://de.wikipedia.org/wiki/Spline-Interpolation

You may remember my "DERIVE program" for investigating a curve from DNL#15. Otto Reichel has written a related DERIVE tool for the "reverse" discussion of a curve. This is the exact translation word by word how we – teachers and students in Austria – often name this kind of problems. In Germany they have another name "Wanted – Problems".

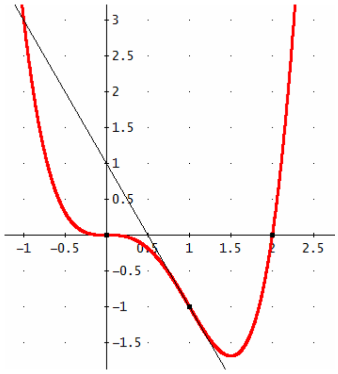
Let´s take a typical problem: The graph of a polynomial function of order 4 passes two points $P(2|0)$ and $Q(-1|0)$. *The origin is an inflection point with 1 as slope of the tangent. Find the equation of the function.*

In my opinion Otto´s file does not need any further explanation. The first method which he uses is very common and not new, but the second one provides a skilled function. DERIVE "looks" for the order of the polynomial automatically and within one step you will obtain the result. This is really a fine tool for teachers and a nice programming technique. Josef

The "Reverse" Discussion of a Curve

Otto Reichel, St. Pölten, Austria

```
                   2
#1:   f2_(x) := a·x  + b·x + c

#2:   f2_1(x) := 2·a·x + b

                    3      2
#3:   f3_0(x) := a·x  + b·x  + c·x + d

                      2
#4:   f3_1(x) := 3·a·x  + 2·b·x + c + d

#5:   Read as: Polynomial of degree 3 - 1st derivative

#6:   f3_2(x) := 6·a·x + 2·b

                    4      3      2
#7:   f4_0(x) := a·x  + b·x  + c·x  + d·x + e

                      3        2
#8:   f4_1(x) := 4·a·x  + 3·b·x  + 2·c·x + d

                       2
#9:   f4_2(x) := 12·a·x  + 6·b·x + 2·c

#10:  Example 1:

#11:  The Graph has two zeros at x1=0 and x2=2, P(1|-1) is an inflection point

#12:  with the slope of the tangent = -2.

#13:  SOLVE([f4_0(0) = 0, f4_0(2) = 0, f4_0(1) = -1, f4_2(1) = 0, f4_1(1) = -2], [a, b, c, d, e])

#14:  [a = 1 ∧ b = -2 ∧ c = 0 ∧ d = 0 ∧ e = 0]

       4      3
#15:  x  - 2·x
```



#19: $\left(\text{SOLUTIONS}([\text{f4\_0}(0) = 0,\ \text{f4\_0}(2) = 0,\ \text{f4\_0}(1) = -1,\ \text{f4\_0}''(1) = 0,\ \text{f4\_0}'(1) = -2],\ [a, b, c, d, e])\right)_1 \cdot \left[x^4, x^3, x^2, x, 1\right]$

#20: $x^4 - 2\cdot x^3$

#21: The general approach is following.

#22: $h(n, x\_, k) := \text{VECTOR}\left(\left(\frac{d}{d\ x\_}\right)^k x\_^j,\ j,\ n,\ 0,\ -1\right)$

#23: $$\begin{bmatrix} h(4, x\_, 0) \\ h(4, x\_, 1) \\ h(4, x\_, 2) \end{bmatrix} = \begin{bmatrix} \left[x\_^4, x\_^3, x\_^2, x\_, 1\right] \\ \left[4\cdot x\_^3, 3\cdot x\_^2, 2\cdot x\_, 1, 0\right] \\ \left[12\cdot x\_^2, 6\cdot x\_, 2, 0, 0\right] \end{bmatrix}$$

#24: $\text{mat}(in) := \text{VECTOR}\left(\lim_{x\_ \to in\downarrow i\downarrow 2} h(\text{DIM}(in) - 1, x\_, in_{i,1}),\ i,\ \text{DIM}(in)\right)$

#25: $\text{coeff}(in) := \text{mat}(in)^{-1} \cdot in\ \text{COL}\ 3$

#26: $\text{func}(in) := \text{coeff}(in)\cdot h(\text{DIM}(in) - 1, x, 0)$

#27: Example 1:

#28: $$\text{matex1} := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 1 & -1 \\ 1 & 1 & -2 \end{bmatrix}$$

#29: $\text{func}(\text{matex1}) = x^4 - 2\cdot x^3$

#30: Example 2:

#31: The graph of a polynomial of degree 4 (quartic) which is symmetric wrt y-axis

#32: has an inflection point in I(-2|3) with slope 4.

#33: $$\text{matex2} := \begin{bmatrix} 0 & -2 & 3 \\ 2 & -2 & 0 \\ 1 & -2 & 4 \\ 0 & 2 & 3 \\ 2 & 2 & 0 \end{bmatrix}$$

#34: $\text{func(matex2)} = \frac{x^4}{16} - \frac{3\cdot x^2}{2} + 8$

#35: $\text{func}\begin{bmatrix} 0 & -2 & 3 \\ 2 & -2 & 0 \\ 1 & -2 & 4 \\ 0 & 2 & 3 \\ 2 & 2 & 0 \\ 1 & 2 & -4 \end{bmatrix} = \frac{x^4}{16} - \frac{3\cdot x^2}{2} + 8$

#36: Example 3:

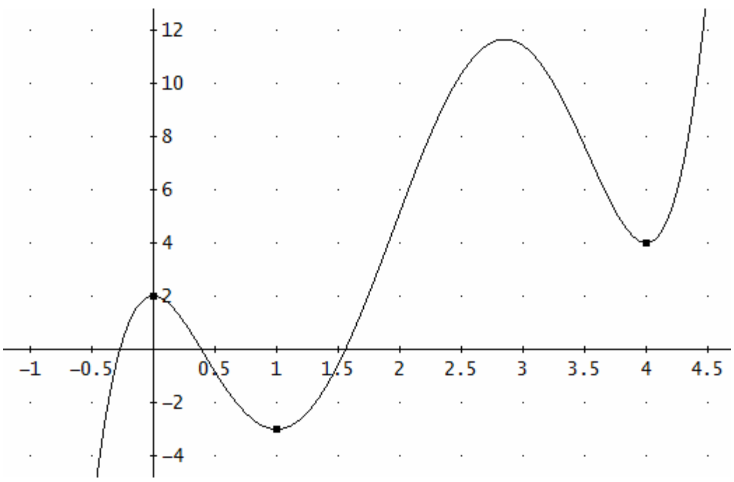
#37: Find the polynomial function with minimum order which shows

#38: three turning points in T1(0|2), T2(1|-3) and T3(4|4).

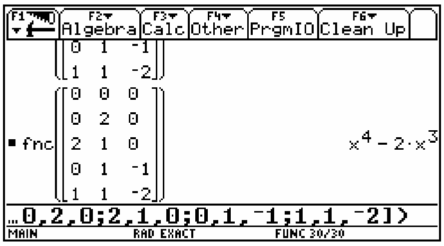
#39: $\text{func}\begin{bmatrix} 0 & 0 & 2 \\ 1 & 0 & 0 \\ 0 & 1 & -3 \\ 1 & 1 & 0 \\ 0 & 4 & 4 \\ 1 & 4 & 0 \end{bmatrix}$

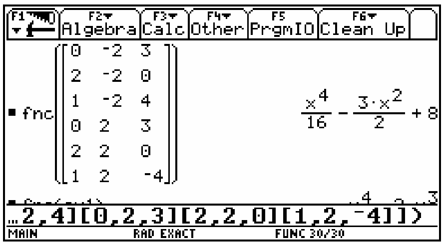
#40: $\frac{313\cdot x^5}{432} - \frac{64\cdot x^4}{9} + \frac{3175\cdot x^3}{144} - \frac{4463\cdot x^2}{216} + 2$

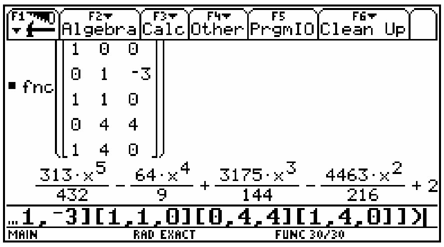
#41: $\begin{bmatrix} 0 & 2 \\ 1 & -3 \\ 4 & 4 \end{bmatrix}$

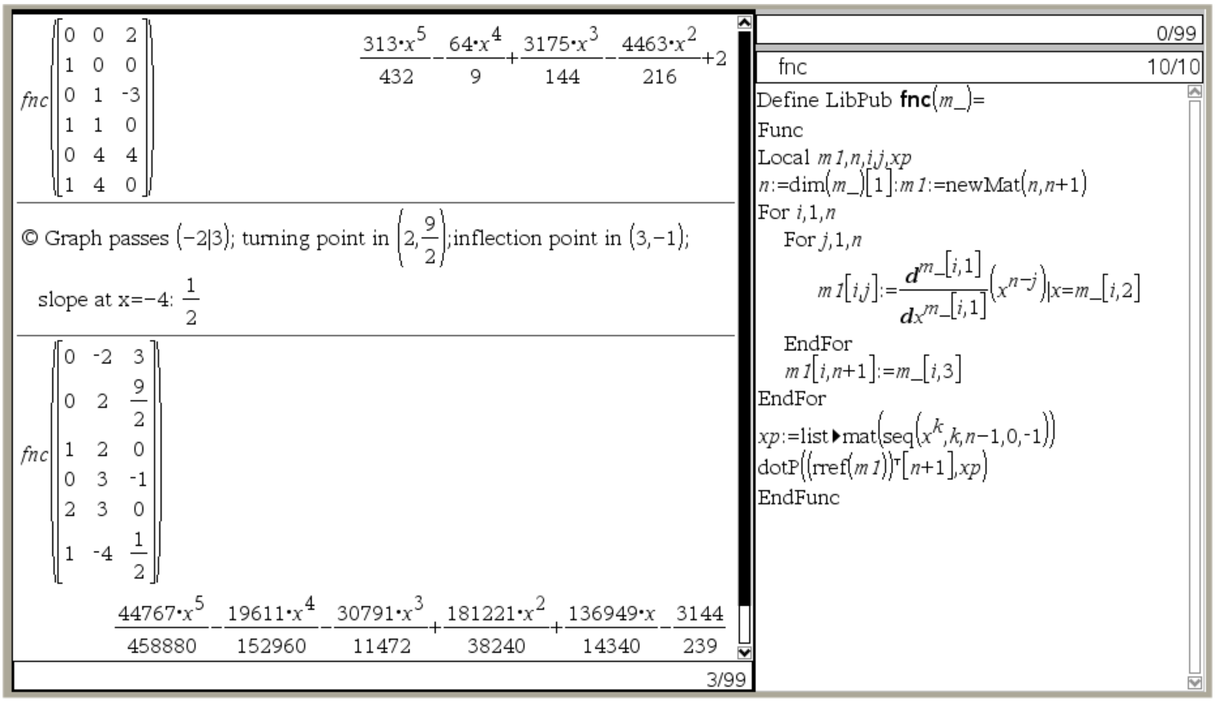


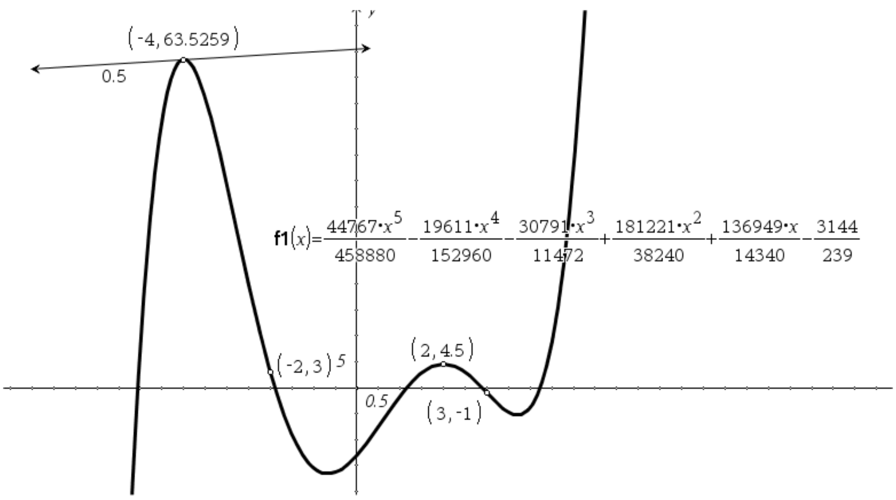
On the next page you can find a ready made tool for finding the polynomial functions for the TI-devices and NspireCAS. The Voyage 200/TI-92 function is the same as the code presented in the TI-Nspire function which follows. There is only one difference: instead of a → b for storing *a* as *b* on the TI-92/V 200 one has to enter b := a with TI-Nspire. Josef


F1 F2▾ Algebra F3▾ Calc F4▾ Other F5 PrgmIO F6▾ Clean Up
■ fnc
x^4 − 2·x^3
...0,2,0;2,1,0;0,1,⁻1;1,1,⁻2])
MAIN RAD EXACT FUNC 30/30


F1 F2▾ Algebra F3▾ Calc F4▾ Other F5 PrgmIO F6▾ Clean Up
■ fnc
x^4/16 − 3·x^2/2 + 8
...2,4][0,2,3][2,2,0][1,2,⁻4]])
MAIN RAD EXACT FUNC 30/30


F1 F2▾ Algebra F3▾ Calc F4▾ Other F5 PrgmIO F6▾ Clean Up
■ fnc
313·x^5/432 − 64·x^4/9 + 3175·x^3/144 − 4463·x^2/216 + 2
...1,⁻3][1,1,0][0,4,4][1,4,0]])
MAIN RAD EXACT FUNC 30/30


fnc
313•x^5/432 − 64•x^4/9 + 3175•x^3/144 − 4463•x^2/216 + 2
© Graph passes (−2|3); turning point in (2, 9/2); inflection point in (3,−1);
slope at x=−4: 1/2
fnc
44767•x^5/458880 − 19611•x^4/152960 − 30791•x^3/11472 + 181221•x^2/38240 + 136949•x/14340 − 3144/239
3/99
0/99
fnc 10/10
Define LibPub fnc(m_)=
Func
Local m1,n,i,j,xp
n:=dim(m_)[1]:m1:=newMat(n,n+1)
For i,1,n
For j,1,n
m1[i,j]:=d^(m_[i,1])/dx^(m_[i,1]) (x^(n−j))|x=m_[i,2]
EndFor
m1[i,n+1]:=m_[i,3]
EndFor
xp:=list▸mat(seq(x^k,k,n−1,0,-1))
dotP((rref(m1))ᵀ[n+1],xp)
EndFunc


(-4, 63.5259)
0.5
y
f1(x)=44767•x^5/458880 − 19611•x^4/152960 − 30791•x^3/11472 + 181221•x^2/38240 + 136949•x/14340 − 3144/239
(-2, 3)
5
(2, 4.5)
0.5
(3, -1)

Titbits from Algebra and Number Theory

by Johann Wiesenbauer, Vienna

In the last article of this series I reported about some minor bugs in version 3.04 of DERIVE. In the meantime all of them have been removed, as I am happy to tell you. Therefore lucky owners of DERIVE 3.05f (or later) should use the updated versions of the following routines in that article (thereby also correcting some misprints of mine):

```
μ(n) ≔ ∏(IF(PRIME(k_), −1, 0), k_, FACTORS(FACTOR(n)))

SQUAREFREE(n) ≔ IF(μ(n)^2 = 1, true, false)

δ(n) ≔ ∏(ITERATE(IF(PRIME(f_^(1/k_)), k_, k_, k_ + 1, k_ + 1),
       k_, 1) + 1, f_, FACTORS(FACTOR(n)))

φ(n) ≔ n·∏(1 − 1/f_^(1/ITERATE(IF(PRIME(f_^(1/k_)), k_, k_ + 1, k_ + 1),
       k_, 1))·f_, FACTORS(FACTOR(n)))

LEGENDRE(a, p) ≔ IF(¬ PRIME(p) ∨ p = 2, "Input error!",
                 MODS(a^((p − 1)/2), p))

PEPINTEST(m) ≔ IF(m = 0, true, IF(MODS(3^2^(2^m − 1), 2^2^m + 1) = −1,
               true, false))
```

In particular, as you may conclude from the last two examples, the MODS-function uses the power-mod-algorithm as well now, when it comes to calculating powers. Great!

Referring to the contribution of Bernhard Wadsack in DNL#17, I would like to point out that DERIVE cannot only display huge Mersenne primes but also prove that they are prime! The basic tool to do this is the so-called Lucas-Lehmer test which can be stated as follows:

> If p is an odd prime, then the Mersenne number $M_p = 2^p - 1$ is prime if and only if $s_{p-1} \equiv 0 \bmod M_p$, where the sequence $s_1, s_2, s_3, \ldots$ is defined by
>
> $$s_1 = 4, \quad s_k \equiv s_{k-1}^2 - 2 \bmod M_p.$$

Its implementation in DERIVE could look like this:

```
LLTEST(p) ≔ IF(p = 2 ∨ ¬ PRIME(p), "p is supposed to be an odd prime",
      IF(ITERATE(MOD(s_^2 − 2, 2^p − 1), s_, 4, p − 2) = 0, true, false))
```

As a first test we will check an assertion of Mersenne in 1644, who claimed that the numbers $M_p = 2^p - 1$ are prime for numbers $p \le 257$ if and only if p is one of the following primes: 2, 3, 5, 7, 13, 17, 19, 31, 67, 127, 257.

For the sake of convenience, we write a little routine that checks all Mersenne numbers $M_p = 2^p - 1$ with $p \le s$ for primality:

```
MERSENNE_LIST(s) ≔ SELECT(LLTEST(p_), p_, SELECT(PRIME(k_), k_, s))
```

The following line was executed in 8.1 seconds (in 1995, 0.172 sec in 2008):

```
MERSENNE_LIST(257) = [3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127]
```

As it turned out, Mersenne made 5 errors by including p = 67, 257 and leaving out 61, 89, 107.

Here are some more examples together with the respective calculation times:

```
LLTEST(521) = true (1.0 sec)          (0.031 with DERIVE 6.10)
LLTEST(607) = true (1.5 sec)          (0.031 with DERIVE 6.10)
LLTEST(1279) = true     (6.7 sec)     (0.11 with DERIVE 6.10)
LLTEST(2203) = true     (28.5 sec)    (0.25 with DERIVE 6.10)
LLTEST(2281) = true     (31.7 sec)    (0.27 with DERIVE 6.10)
LLTEST(3217) = true     (83.1 sec)    (0.50 with DERIVE 6.10)
LLTEST(4253) = true     (183.4 sec)   (0.97 with DERIVE 6.10)
LLTEST(4423) = true     (207.8 sec)   (1.06 with DERIVE 6.10)
LLTEST(9689) = true     (2092.6 sec)  (9.94 with DERIVE 6.10)
```

Though the calculation times are increasing very rapidly, the primality testing of Mersenne numbers with several thousand digits is well within reach! The reader might try some more exponents of the following list, yielding Mersenne primes, namely

p = 9941, 11213, 19937, 21701, 23209, 44497, 86243, 110503
132049, 216091, 756839, 859433.

(Well, to be honest, I wouldn´t recommend taking numbers of the second half of this list!)

We turn to a more interesting question, namely: How can we find actual factors of a Mersenne number which has proven to be composed by the Lucas-Lehmer test? In general, this is a very difficult question.

Let´s take as an example the case p = 67 from above where Mersenne was mistaken in believing that it yields a Mersenne prime. Applying factor to $M_{67} - 1$ yields in 1.8 sec (!) the following decomposition

```
FACTOR(2^67 - 1) = 193707721·761838257287   (now in 0.078 sec!)
```

(By the way, F. N. Cole gave a talk about this factorization at a meeting of the AMS in 1903; according to his own words it took him the "Sundays of three years" to find it.) The following calculations show that both factors are ≡ 1 mod 2p and ≡ ±1 mod 8, which is true in general:

$$\text{FACTOR}(193707721 - 1) = 2^3 \cdot 3^3 \cdot 5 \cdot 67 \cdot 2677$$

$$\text{FACTOR}(761838257287 - 1) = 2 \cdot 3^2 \cdot 29 \cdot 67 \cdot 2551 \cdot 8539$$

$$\text{MODS}(193707721,\ 8) = 1$$

$$\text{MODS}(761838257287,\ 8) = -1$$

This fact could be used to speed up trial division. In case of the first prime factor q we also notice that $q - 1$ splits up into small prime factors apart from a single bigger one. There is a factorization method introduced by Pollard called (p – 1)-method which takes advantage exactly of this fact.

Suppose that for a given number n and some boundaries $s1$, $s2$ all prime powers contained in n are less than $s1$ apart from a single prime which must be less than $s2$. By forming appropriate powers a^r of an integer $a > 1$ and calculating the GCD($a^r - 1$,n) eventually a factor of n will be found. The interested reader will have no difficulty in finding out the details by looking into the following DERIVE-routine.

```
PMINUS1(n, a, s1, s2) := (ITERATE(IF(c_ > s2 OR e_ > 1, [a_, b_, c_, d_, e_],
   [MOD(a_·MOD(b_^(d_ - c_), n), n), b_, d_, NEXT_PRIME(d_), GCD(a_ - 1, n)]),
   [a_, b_, c_, d_, e_], ITERATE(IF(c_ > s1 OR e_ > 1, [a_, b_, c_, d_, e_],
   [IF(d_^2 > s1, MOD(a_^d_, n), MOD(a_^d_^FLOOR(LN(s1), LN(d_)), n)), a_, d_,
   NEXT_PRIME(d_), GCD(a_ - 1, n)]), [a_, b_, c_, d_, e_],
   [a, 1, 1, 2, GCD(a, n)])))↓5
```

If n is of the form $(2^p - 1)/f$, where f is some already known factor, we should start with a^p instead of a. This is an adoption of the (p – 1)-method taking this into account:

```
MPMINUS1(p, f, a, s1, s2) := PMINUS1((2^p - 1)/f, a^p, s1, s2)
```

Now let´s try if it works:

```
MPMINUS1(67, 1, 30, 50, 10000) = 193707721
```
(2.7 sec) (0.11 sec now)

Terrific! This execution time comes even close to the time of the built-in factor-routine which is already amazingly good!

The (p – 1)-method works also for Fermat numbers $F_m = 2^{2^m} + 1$ which have turned out to be composed by Pépin´s test. The other parameters in the following routine FPMINUS1 have the same meaning as in MPMINUS1:

$$\mathrm{FPMINUS1}(m, f, a, s1, s2) := \mathrm{PMINUS1}\left(\frac{2^{2^m} + 1}{f}, a^{2^{m+2}}, s1, s2\right)$$

```
FPMINUS1(10, 1, 3, 40, 100) = 6487031809
```
(0.031 sec)

$$\mathrm{FACTOR}(6487031809 - 1) = 2^{14}\cdot 3^2\cdot 29\cdot 37\cdot 41$$

For Fermat numbers $F_m = 2^{2^m} + 1$ every factor is of the form $k\,2^{m+2} + 1$ for some k. The following routines calculate factors of F_m and the corresponding k.

```
LDF(m, s1, s2) := ITERATE(IF(MOD(2^2^m, t_) = t_ - 1, t_, IF(t_ > s2, 1,
                  t_ + 2^(m + 2))), t_, IF(MOD(s1, 2^(m + 2)) = 1, s1,
                  (FLOOR(s1 - 1, 2^(m + 2)) + 1)·2^(m + 2) + 1))

KLDF(m, s1, s2) := (LDF(m, s1·2^(m + 2) + 1, s2·2^(m + 2) + 1) - 1)/2^(m + 2)

KLDF(1945, 1, 10) = 5
```
(0.48 sec)

The Fermat number F_{1945} can never be seen in full length because the number of particles in our universe would not suffice to print it, as Coxeter pointed out. Even so, we have just calculated the factor $5 \cdot 2^{1947} + 1$ of it! More about Fermat numbers and their factors another time!