THE DERIVE - NEWSLETTER #9

THE BULLETIN OF THE

DERIVE®

USER GROUP

C o n t e n t s:

revised reprint 2006 **March 1993**

Our member from the first days, Herbert Appel asks for publishing the following announcement. I hope I can help you and much luck, Herbert.

Some food for your memory: DERIVE XM

DERIVE XM operates identically to the regular DERIVE but can take advantage of up to 4 gigabytes (4 billion bytes) of extended memory. The size of problems that DERIVE XM can handle is now limited only by your computers's memory and speed.

Operations such as expanding polynomials, inverting matrices, and solving sysems of linear equations can require a great deal of memory to store the intermediate and/or final results. For example, the maximum size random number matrix that regular DERIVE can invert is about 40 by 40. However, DERIVE XM was able to invert a 100 by 100 matrix an a computer with 4 MB of total memory (i.e. of extended memory).

However, the system requirements for DERIVE XM are still relatively modest: a 386 or 486 based PC compatible computer with a minimum of 2 MB of memory (i.e. 1 MB of conventional and 1 MB of extended memory).

Use MathPORT to Drive DERIVE to the Publisher:

With DERIVE and MathPORT you not only get the right answer, but you can share your professional-quality documents with others! MathPORT is the leader in converting math equations for word processed documents.

The equations beneath were created and ublished as TIFF graphics files by MathPORT. They were automatically converted from DERIVE files without editing.

(K-Talk Communications
30 West First Ave. Suite 100
Columbus, OH 43201)

Liebe DERIVE Anwender!

Ich möchte Sie alle zu Beginn des 3. DNL-Jahres recht herzlich begrüßen und Ihnen auf diesem Weg für die vielen lieben Grüße anlässlich des Jahreswechsels danken. Die breite Zustimmung für die Ge-staltung des DNL hat mich sehr gefreut. Viele von Ihnen haben den kleinen Fragebogen aus DNL#8 ausgefüllt und so ist ein ganzer Berg von Anregungen zusammengekommen. Da noch immer Rückmeldungen einlangen, möchte ich erst im nächsten DNL eine Zusammenstellung Ihrer Ideen und Anregungen veröffentlichen.

Natürlich möchten einige von Ihnen mehr schulbezogene Themen, andere befürchten eine allzu große Schullastigkeit. Einige sind Grafikliebhaber - so wie ich, ich kann das nicht verleugnen - andere empfinden manche Dinge als Spielerei. So werde ich nun versuchen den Wünschen halbwegs gerecht zu werden.

Eine besonders nette Geschichte hat der Artikel von Marko Horbatsch: Ein Grazer DUG Mitglied schickte mir eine Kopie, darauf hin nahm ich in meinem „besten Englisch“ Kontakt mit M.Horbatsch auf und bekam sehr rasch einen netten Brief in . . . . . bestem Deutsch zurück. Herzlichen Dank Herr Horbatsch für Ihre Erlaubnis zur Veröffentlichung.

Für den nächsten DNL werde ich einem häufig genannten Wunsch nachkommen und den Schwerpunkt Statistik und Wahrscheinlichkeitsrechnung setzen. Vielleicht hat jemand von Ihnen einen Beitrag zu diesem Thema auf Lager?

Vielfach wurde der Wunsch nach einer Information über die Möglichkeiten der neuesten DERIVE Versionen geäußert. Sie können auf Seite 41 eine Zusammenstellung der wichtigsten neuen Features finden. (2.50 und höher) Beachten Sie auch den kurzen Bericht in den benachbarten Informationen über DERIVE XM.

Aufgrund der - an sich erfreulichen - Materialfülle muss ich Sie bitten, für dieses Mal auf den BBS zu verzichten. Auch das User Forum wurde sehr kurz gehalten. Einige umfangreiche Briefe warten auf ihre Veröffentlichung.

Abschließend bitte ich alle jene, die ihren Mitgliedsbeitrag für 1993 noch nicht entrichtet haben, dies bei nächster Gelegenheit nachzuholen. Herzlichen Dank.

Bis zum nächsten Mal mit den besten Grüßen Ihr

Dear DERIVE Users,

First I want to welcome you to the start of DUG's third year of existence. I also want to thank everybody for the nice greetings and for the many good wishes I and my wife received for the New Year. Many of you have sent back the little form from DNL#8 and so a lot of ideas have assembled in my DNL-folder. So we are receiving answers every day I want to give a list of the ideas, suggestions and criticisms in the next DNL.

Of course some of you want to have more contributions connected with didactics and teaching mathematics while others may be afraid of too much schoolmathematics, they would prefer more scientific applications. Some like graphics - as I do, I can't deny it - others dislike them as a triviality. I'm trying to find a way to satisfy most of your interests

Marko Horbatsch's "Physics in the Classroom" has an especially nice story. A former DUG member from Graz, A, sent me a copy and I contacted M.Horbatsch in Canada in my 'best English' and very soon after I received a friendly letter . . . . but in "best German"! Many thanks Marko for giving your permission to reprint your interesting work.

In the next DNL I will fulfill a wish mentioned from many members and concentrate on statistics and probability theory. Maybe there are some readers who have a DERIVE-application on these subjects?

Widely I've noted the wish for information about the facilities of DERIVE's latest versions. On page 41 you will find a list of the most important new features (2.50 and higher). Please note the comment on DERIVEÿXM on the 'Information page'.

Because of the - really delightful - quantity of material received I must ask you to do without the BBS this issue. The User Forum had to be shortened, too. Some extensive letters are waiting for publication.

Finally I'd like to ask the DUG-Members who haven't paid their membership fees for 1993 to do this at their earliest opportunity. Thank you.

Until next time
with my best regards

The *Derive-News-Letter* is the Bulletin of the Derive-User Group. It is published at least three 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 didactic manners in teaching Mathematics.

Editor: Mag. Josef Böhm
A-3042 Würmla
D´Lust 1
Austria
Phone/FAX:43-(0)2275/8207
e-mail: nojo.boehm@pgv.at

Contributions:

Please send all contributions to the above address. 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 will be warmly welcomed nonethe-less. Your contributions will be edited but not assessed. By submitting articles the author gives his consent for reprinting it in the *D-N-L*. The more contributions you will send to the Editor, the more lively and richer in contents the Derive-News-Letter will be.

Preview: (Contributions for the next issues)

Census Assigment, Some Worksheets, K. Eames, UK
Bisection with *DERIVE*, D.M. Dyer, USA
Fluid Flow in *DERIVE*, Reuther a.o., BRA
Newton´s Chaos, Graphic Integration, J. Böhm, AUT
Computer Aided Mathematics in School, K.H. Keunecke, GER
Continued Fractions, R. Setif, FRA
DERIVE in Hawaiian Classrooms, Sawada, USA
Stability of ODEs, Kozubik, SLK
Bivariate Normal Distribution, Marinell, AUT

DNL#10 will be published June 93

Download all *DNL-DERIVE*- and TI-files from
http://www.austromath.ac.at/dug/
http://www.bk-teachware.com/main.asp?session=375059

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

Reuther & Domingos, Petropolis, Brazil

We would like to submit our preprint "Extended Fourier Series with DERIVE" to the DNL. We hope it will fulfill all the specifications.
We would also like to remember that this is our second participation, because a member of our group, Patricia R. Medici, has been in the congress happened in Krems in April. Our group is a small one, and we're trying to widespread the DERIVE "culture" in Brazil.

K.Eames, Kingsway College, UK

On receiving the recent copy of DNL I thought you might like to know how DERIVE is being used at Kingsway College, which is a Further Education College. The reasons why DERIVE is being looked at are as follows:

1. Kingsway is interested in the impact of such packages on the Mathematics curriculum, I felt DERIVE is better for the 16-19 age group than REDUCE or MATHEMATICA, the former seemed to be too complicated and the latter too memory hungry.

2. Kingsway is already producing manuals and assignments for A-level and University of North London students to try out on their respective courses. Student manuals on DERIVE's Basic commands and Use of DERIVE in Algebra complete with simple test exercises for students to complete, in order for them to see if they have mastered the relevant commands. There are also manuals on DERIVE's graphical capabilities along with worksheets and on DERIVE's use in Calculus.

3. Kingsway is currently looking at the possible use of DERIVE as a means of introducing certain topics, Matrices for example.

plus I have given a number of INSET sessions on the Use of DERIVE for the London Branch of the Mathematical Association and for Suffolk College of Higher Education. A copy of the INSET manual is enclosed. Many of the sheets are being updated at this present moment in time.

My own views on the use of DERIVE are as follows:

1. In supporting investigative work, thereby enabling more time to be spent on formulating problems and predecting possible results

2. To introduce topics

3. To enhance students opportunities in their course and to come into contact with it before going onto HE

4. To check results to problems solved without the use of DERIVE.

K.de Ridder, Zwolle, NL

Wij zijn met een experiment bezig om computeralgebra te introduceren oop de H.T.S. Zwolle, studierichting elektrotechniek. Dat experiment begint februari 1993. Het gebruik van CA als wiskundig gereedschap vor het oplossen van technische problemen staat darbij voorop.(the use of CA as a mathematical tool to solve technical problems is the most important aim). Als U geinteresseerd bent willen we U graag op de hoogte houden van onze vorderingen en resultaten.

DNL: I think we would be interested in your experiments and we hope to hear from you when you have got any results.

There are some other extensive letters - from MrÿNeurath, Mr Pröpper, Mr Schmidt, Mr Williams, Mr Horbatsch and others - which I want to publish in the next issue. Many thanks for your interest and cooperation.

Solving Physics Problems in the Classroom with *DERIVE* 2.0

Marko Horbatsch, York, Ontario, Canada

Abstract: We present an overview of problems that can be solved now in *DERIVE* 2.0 due to the incorporation of routines that handle iteration as well as other programming elements such as logical branching. The applications include recursive calculation of orthogonal polynomials, solution of iterative maps, numerical solution of ordinary and simple partial differential equations.

DERIVE® A Mathematical Assistant [1] has proven to be a useful educational as well as research tool [2-4]. With the new version 2.0 some of the major constructive criticisms raised in our previous review have been taken into account. The basic philosophy of the program reamins the same, namely to achieve compact code that can be loaded on almost any MS-DOS based machine irrespective of the CPU. This is achieved by defining intrinsically to the program only a minimal set of functions and relegating all special functions to external files. The improvements to the package are in two areas. On the program side one has added to the symbolic environment two important aspects of a programming language, namely conditional branching as well as iteration. On the support files new files with function definitions were added and a description of these is provided in a much enhanced manual [1]. Also the intrinsic help utility covers now the utility files in a brief fashion so that it is possible to use them without the manual. Extensions to the already ample graphing capabilities of the program include the possibility to draw user calculated isometric projections of three-dimensional curves, surfaces and vectors in two-dimensional plot windows as well as graphing of complex functions.

The purpose of this review is to demonstrate how Derive´s new features can be combined with the graphical capabilities to extend significantly the program´s usefulness for educational and research purposes. It will be shown using such examples as dicrete maps, solution of ordinary and simple partial differential equations that it is possible to set up a theoretical physics laboratory with this environment.

A major problem for the instructor of a course in modern physics in an undergraduate curriculum is to familiarize the student with the idea of a differential equation – eigenvalue problem. Instead of concentrating on the physics behind *Schrödinger*´s equation the student is preoccupied with learning the mathematics. The symbolic-graphic environment that Derive provides is ideally suited to present results and allow self-study of differential equations, orthogonal polynomials, their recursion relations, etc.

Special software packages are in existence for purpose of plotting potentials and the eigenvalues and eigenfunctions of simple Hamiltonians. The following example will show ow the same can be achieved with a few programming steps in Derive. An advantage over dedicated packages is that the student has to understand how Derive managed to obtain these results.

In detail: Line #1 in fig. 1a provides a definition for the Hermite polynomials, which is included in one of the support files that come with the software. Derive provides access to many special functions in this way, i.e. one can use after importing such a line of code into memory. While this may seem awkward compared to programming languages as MAPLE, where such functions can be called as intrinsic functions, for educational purposes, one has the advantage of seeing a definition of the function displayed. In the new version of Derive utility files can also be loaded into memory without displaying them as a part of the user´s program.

#1: $\mathrm{HERMITE\_H}(n, x) := (-1)^n \cdot e^{x^2} \cdot \left(\frac{d}{dx}\right)^n e^{-x^2}$

#2: $\mathrm{PSI}(x, n) := \mathrm{EXP}(-0.5 \cdot x^2) \cdot \mathrm{HERMITE\_H}(n, x) \cdot \sqrt{\left(\frac{1}{\sqrt{\pi} \cdot 2^n \cdot n!}\right)}$

#3: $\mathrm{HAMILTON}(u) := 0.5 \cdot \left(-\left(\frac{d}{dx}\right)^2 u + x^2 \cdot u\right)$

#4: $\mathrm{ev} := \mathrm{VECTOR}\left(\frac{\mathrm{HAMILTON}(\mathrm{PSI}(x, i))}{\mathrm{PSI}(x, i)}, i, 0, 3\right)$

#5: $\mathrm{ev} := \left[\frac{1}{2}, \frac{3}{2}, \frac{5}{2}, \frac{7}{2}\right]$

#6: $\mathrm{VECTOR}(0.5 \cdot \mathrm{PSI}(x, k - 1) + \mathrm{ELEMENT}(\mathrm{ev}, k), k, 1, 4)$

Figure 1a: A calculation of the lowest four eigenvalues and eigenfunctions of the harmonic oscillator (in units *h* = *m* = *w* = 1). In line #4 DERIVE calculates the eigenvalues with results shown in line #5. Line 6 provides a vector in which the lowest four eigenfunctions shifted by the corresponding eigenvalues are generated .

Line #2 defines the n^{th} harmonic oscillator eigenfunction with appropriate normalization factor. As Derive has no intrinsic knowledge of the Hermite polynomials and their properties, it is not able to prove in general that the (real) functions defined in this line satisfy the orthonormality condition

$$\int_{-\infty}^{\infty} \psi_n(x)\psi_m(x)dx = \delta_{nm} . \qquad (1)$$

It is, however, possible to verify this property for individual cases of *m* and *n* and this is calculated exactly in Derive.

Line #3 provides the Hamiltonian as it acts on some arbitrary function *u*, which for our choice of potential and units reads:

$$H = -\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}x^2 . \qquad (2)$$

Note that the operation with the Hamiltonian operator on *u* is defined as a function with *u* as an argument., but that this function includes the action of a differential operator on *u*. Line #4 calculates a vector that contains the lowest four eigenenergies form the equation $H\psi = E\psi$, with the result displayed in line #5. For plotting purposes a vector is generated whose elements contain the lowest four eigenfunctions shifted from each other by the eigenvalues. This is done in order to show in fig. 1b the spatial extent in which a classical one-dimensional harmonic oscillator would move, if it was given the corresponding amount of energy.

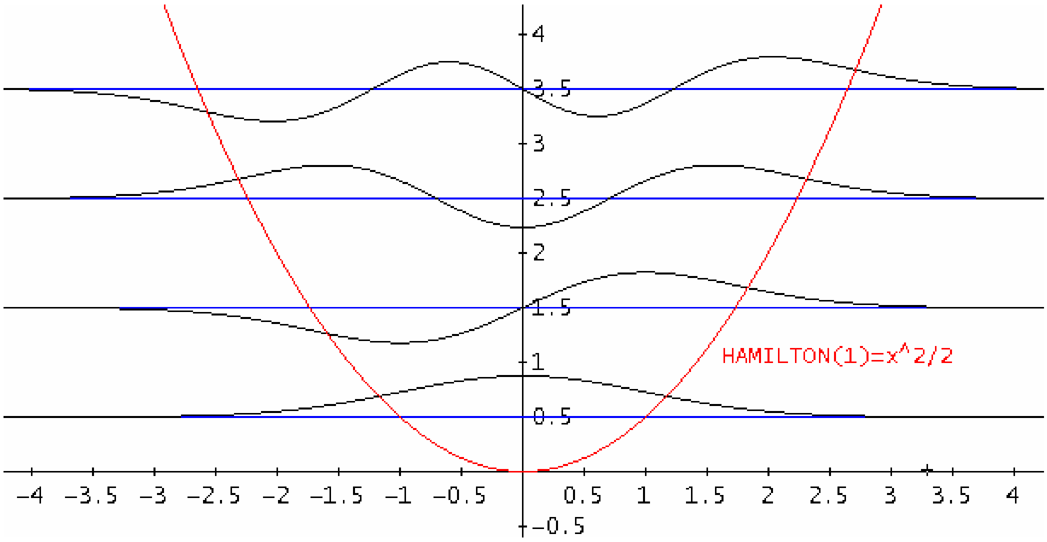


Figure 1b: The eigenfunctions displayed in a Derive 2D-Plot Window

By taking the square of the eigenfunctions one could show on a similar graph the probability of finding a quantum mechanical particle at a given location *x* (assuming that it is in a given eigenstate *n*) and draw attention to the oscillatory structure of this quantity as well as the classically forbidden region.

One could argue that such a figure is included in many textbooks on modern physics or quantum mechanics and ask whether a student will learn much from seeing the same on a computer screen. Furthermore, the essentially effortless calculation summarized by four lines of Derive code could discourage students from performing these calculations by hand. An appropriate way to make use of Derive for students who are supposed to master the mathematics involved, would be in such a way that the program would be used for checking and gaining confidence in a calculation by hand. Solutions to radial *Schrödinger* equations for several commonly used potentials involving other orthogonal polynomials can be obtained in the same way.

The calculations shown so far were feasible in previous versions of Derive. A novel feature is the possibility to study recursion relations for symbolic expressions, one of the strong points of symbolic languages. The line of Derive code given below provides an alternative calculation of the Hermite polynomials based upon the recursion relation

$$H_{M+1}(x) = 2xH_M(x) - 2M\,H_{M-1}(x), \qquad (3)$$

which is started by $H_0(x) = 1$ and $H_1(x) = 2x$.

The new function in Derive 2.0, which generateds recursions is ITERATE. Its syntax is given by ITERATE($f(w)$, w, w_0, n) and it returns the n^{th} iteration of the assignment $w \to f(w)$ starting with w_0. A similarly working function ITERATES returns all elements of the iteration. The first three arguments are allowed to be vectors, which in our case enables the two-step recursion to be calculated. In our example the symbolic placeholder *w* is a three-vector as we have to update the iteration index in addition to storing $H_M(x)$ and $H_{M-1}(x)$. Reference to the individual elements in the recursion is made by Derive´s routine to extract elements from vectors (and matrices), ELEMENT (v,j) returns the j^{th} element of vector *v*. Thus the recursion in (1) is programmed in Derive by the following line to generate $H_n(x)$.

```
ELEMENT(ITERATE([ELEMENT(w, 2), 2*x*ELEMENT(w, 2) -
2*(ELEMENT(w, 3) + 1)*ELEMENT(w, 1), ELEMENT(w, 3) + 1], w,
[1, 2*x, 0], n), 1)
```

Here the first element of w returns $H_n(x)$, the second stores $H_{n-1}(x)$ and the third element of w contains the index n. The square brackets in Derive code refer to a vector given in terms of its elements. The fact that one has to update the iteration index and refer to it by the ELEMENT-function is clumsy and this should be simplified in some future release. Once the student has learned how to set up recursions in Derive, he can save a lot of time by avoiding tedious calculations, compare results with the definition given in line #1 of fig. 1a and concentrate on the content rather than the technical aspects of the calculation.

$$\left(\mathrm{ITERATE}\left(\left[w_2,\ 2\cdot x\cdot w_2 - 2\cdot(w_3 + 1)\cdot w_1,\ w_3 + 1\right],\ w,\ [1,\ 2\cdot x,\ 0],\ n\right)\right)_1$$

Recent Derive versions don´t need the ´clumsy´ `ELEMENT`-function

This new capability of handling iteration opens up possibilities held traditionally by numerical/graphical environments, such as Quickbasic or Turbo-Pascal. Derive is easier to program than these environments and has an easier access to graphing in multiple windows. We have chosen an example from classical mechanics that cannot be solved in closed form. Its intent is to illustrate possible trajectories for a classical charged particle with small mass that impinges on a two dimensional fixed array of oppositely charged ions.

The ITERATES function is used to integrate *Newton´s* equation for a particle moving through an array of point-charges. The dimensionless equation ($m = e = 1$) is solved approximately by the so-called leapfrog algorithm coded in line #11 below:

$$\mathbf{v}(t_{n+1/2}) = \mathbf{v}(t_{n-1/2}) + dt\,\mathbf{F}(\mathbf{r}(t_n)),\quad \mathbf{r}(t_{n+1}) = \mathbf{r}(t_n) + dt\,\mathbf{v}(t_{n+1/2}). \qquad (4)$$

Here $\mathbf{r}$, $\mathbf{v}$ and $\mathbf{F}$ are the position, velocity and force vectors respectively and $\mathbf{r}(t_0)$ and $\mathbf{v}(t_{-1/2})$ the initial conditions. This fixed time-step algorithm is not very accurate but economical for many applications. In order to avoid accuracy problems, which would force one to use a very small time step dt, the potential of the static sources has been regulated: while the motion of the test charge is confined to the x-y plane, the external sources can be thought of as resinding in a plane located at a distance $x = \sqrt{a}$ above (or below). The solution generated by `LEAP(n)` is a vector variable named `vek`, which contains the x- and y-coordinates of $\mathbf{r}(t_0)$ and $\mathbf{v}(t_{n+1/2})$. One can see that the coding is somewhat clumsy due to the fact that one has to iterate on a single object and refer to its elements. The code shown below would appear less intimidating, if we had changed the ELEMENT function to one with a shorter name.

The routine EXTRACT_2_COLUMNS is supplied in one of the support files and serves to extract two colums out of a given matrix, which is achieved by transposing the matrix, extracting two rows and transposing again (lines #1 and #2). `xvek` and `yvek` are vectors denoting the x- and y-coordinates of the points at which the fixed charges are located *ncent* calculates the number of these charges. `areg` is used to regulate the interaction and lines #7 and #8 provide the x- and y-components of the force acting on a test-charge located at *(x_, y_)*. `incond` specifies the initial condition $\mathbf{r}_0 = (2,0.4)$ and $\mathbf{v}_0 = (-0.5,0)$ of the test-charge in the plane. As the ITERATES function returns a matrix containing positions and velocities on the time mesh one has to the columns containing the (x,y)-values in order to graph the trajectory.

```
#1:   EXTRACT_2_ELEMENTS(v, p, q) := [ELEMENT(v, p), ELEMENT(v, q)]

#2:   EXTRACT_2_COLUMNS(a, c1, c2) := EXTRACT_2_ELEMENTS(a', c1, c2)'

#3:   xvek := [-1, 0, 1, -1, 0, 1, -1, 0, 1]

#4:   yvek := [0, 0, 0, 1, 1, 1, -1, -1, -1]

#5:   ncent := DIMENSION(xvek)

#6:   areg := 0.5

                   ncent                     x_ - ELEMENT(xvek, i)
      FX(x_, y_) :=  Σ   ─────────────────────────────────────────────────────────────
#7:                 i=1                        2                          2        3
                        √((x_ - ELEMENT(xvek, i))  + (y_ - ELEMENT(yvek, i))  + areg)

                   ncent                     y_ - ELEMENT(yvek, i)
      FY(x_, y_) :=  Σ   ─────────────────────────────────────────────────────────────
#8:                 i=1                        2                          2        3
                        √((x_ - ELEMENT(xvek, i))  + (y_ - ELEMENT(yvek, i))  + areg)

#9:   incond := [2, -0.5, 0.4, 0]

#10:  dt := 0.01

#11:  LEAP(n) := ITERATES([ELEMENT(vek, 1) + dt·ELEMENT(vek, 2), ELEMENT(vek, 2) -

        dt·FX(ELEMENT(vek, 1), ELEMENT(vek, 3)), ELEMENT(vek, 3) + dt·ELEMENT(vek, 4),

        ELEMENT(vek, 4) - dt·FY(ELEMENT(vek, 1), ELEMENT(vek, 3))], vek, incond, n)

#12:  EXTRACT_2_COLUMNS(LEAP(1200), 1, 3)
```

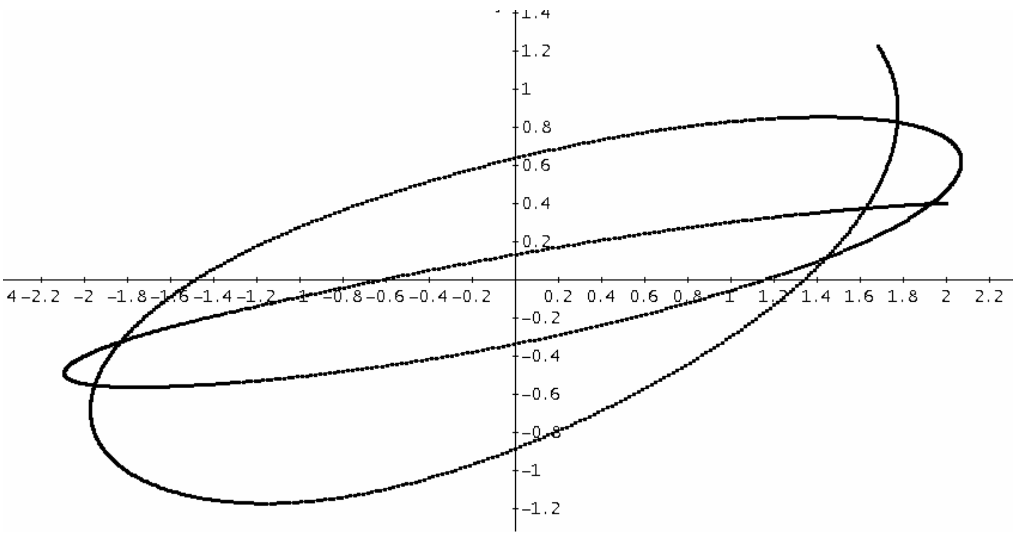


Figure 2: Numerical solution of *Newton´s* equation for a test charge impinging on an array of fixed charges located in a plane with the leapfrog algorithm.
(calculated with DERIVE XM within 1347 sec; DERIVE 2.54 wouldn´t do it in one run, I had to split. DERIVE 6.10 needs 7 sec in 2006. Editor)

See the leapfrog algorithm – performed with DERIVE 6 – in the following. The output looks the same!

$$\#1: \quad pts := \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 1 & 0 \\ -1 & 1 \\ 0 & 1 \\ 1 & 1 \\ -1 & -1 \\ 0 & -1 \\ 1 & -1 \end{bmatrix}$$

#2: [areg := 0.5, incond := [2, -0.5, 0.4, 0], dt := 0.01, vek :=]

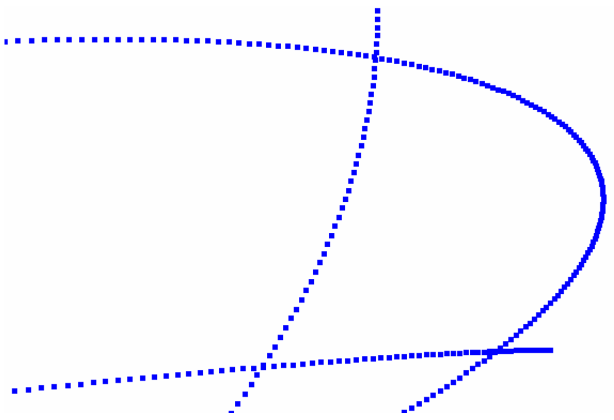
$$\#3: \quad FX\_(x\_, y\_) := \sum_{i=1}^{DIM(pts)} \frac{x\_ - pts_{i,1}}{\sqrt{\left((x\_ - pts_{i,1})^2 + (y\_ - pts_{i,2})^2 + areg\right)^3}}$$

$$\#4: \quad FY\_(x\_, y\_) := \sum_{i=1}^{DIM(pts)} \frac{y\_ - pts_{i,2}}{\sqrt{\left((x\_ - pts_{i,1})^2 + (y\_ - pts_{i,2})^2 + areg\right)^3}}$$

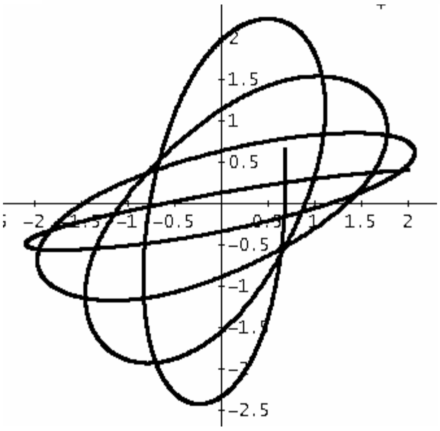
$$\#5: \quad LEAP\_(n) := ITERATES\left(\left[vek_1 + dt \cdot vek_2,\ vek_2 - dt \cdot FX\_(vek_1, vek_3),\ vek_3 + dt \cdot vek_4,\ vek_4 - dt \cdot FY\_(vek_1, vek_3)\right], vek, incond, n\right)$$

#6: (LEAP_(1200)) COL [1, 3]

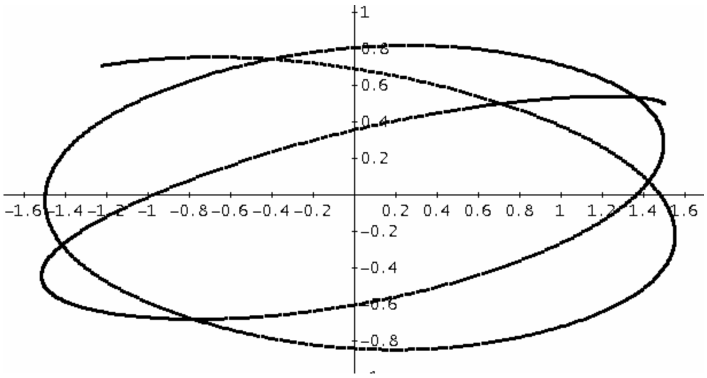
The results shown in fig. 2 are produced upon the simplification of line #12 and displayed as a sequence of points. These points indicate the position of the moving charge at discrete time-steps. Subsequenent points are displayed in different colors chosen from a palette on the computer screen and could be represented in the printout by different grey shades. From the graph one obtains a feeling for the speed of the test charge in the various parts of the trajectory. The spacing of the points indicates where the probe charge accelerated and decelerated, etc.. Further interesting quantities could be calculated and displayed as function of time. One can easily study the sensitivity of such a system on the initial conditions and watch how a minimal change in the initial position or velocity makes the test charge take a different turn at some point during its path.



(LEAP_(2400)) COL [1, 3]



incond := [1.5, -0.3, 0.5, 0.2]



A classical mechanics laboratory can be set up following this example, i.e., one can assemble a number of problems with different potentials of relevance to physics, chemistry, etc., for numerical study. An interesting comparison comes to mind with existing numerical software packages dedicated to such problems, such as Chaos Demonstrations (published by the American Physical Society), Chaos Dynamics Workbench (APS), Phaser (Springer) and others. On this front Derive cannot compete at all in terms of speed, as all computations are done without a coprocessor and usually in exact precision (rational) arithmetic. Another technical problem is the fact that all values of the iteration are stored and extracted for plotting only subsequently. If one is forced to a small time-step for the integration, this may lead to storage problems due to the DOS 640 kB memory barrier. However, from an educational point of view Derive is preferable, as, again, one is forced to understand what is going on. For the production of nice figures, where large amounts of data and plotting on the run are required, one could then turn to one of the programs mentioned above.

At this point it should be clear that Derive provides an interesting tool for the numerical analyst (as long as one is content with basic problems, such as occur in the classroom). Without having to deal with round-off errors (if the problem is solvable in the rational numbers) one can study the effects of dicretization errors, and display results conveniently. However, it is practically impossible to write complicated programs, as Derive still cannot be called a programming language. (Things have changed, Ed.) While many elements of looping, conditionals, iteration are now implemented, one has to live with the structure of single line calls. To code a complicated branching expression is possible, but such a single line can become quite long and very difficult to debug. (I repeat: Things have changed, Ed.)

There exist examples for which the VECTOR function which covers one element of looping, will not simplify its main argument. I found it impossible, e.g., to generate a sequence whose elements were determined by the approximate solution of a nonlinear equation, even though individually Derive´s SOLVE function calculated each element very quickly. From the point of view of symbolic language syntax Derive´s relationship to a complete language (such as Maple, Reduce, etc.) can be compared to that of early Basic dialects to Fortran. On the other hand it is so much easier to use than any of the fullblown symbolic programming languages known to me that it should definitely be used as an introduction to symbolic computing.

Marko Horbatsch, Dept of Physics & Applied and & Computational Mathematical Science
Programme, York University, Toronto, Ontario, M3J 1P3, Canada

Maybe that there are few oy you who – like me – do not remember much about Orthogonal Polynomials. I tried to find some interesting facts about *Hermite Polynomials*, Josef.

About Hermite Polynomials

There is a wide group of so called *Orthogonal Polynomials*. Among others we find *Chebishev, Legendre, Laguerre* and *Hermite Polynomials*. They all have one important common property:

$$\int_a^b p_n(x)p_m(x)\omega(x)\,\mathrm{d}x = \begin{cases} 0 & \text{for } n \neq m \\ \neq 0 & \text{for } n = m \end{cases}$$; ω(x) is the *weight function*.

Chebyshev Polynomials are implemented in DERIVE:

#1: $\text{CHEBYCHEV\_T}(3, x) = 4\cdot x^3 - 3\cdot x$

#2: $\text{CHEBYCHEV\_T}(6, x) = 32\cdot x^6 - 48\cdot x^4 + 18\cdot x^2 - 1$

#3: $$\int_{-1}^{1} \frac{\text{CHEBYCHEV\_T}(3, x)\cdot\text{CHEBYCHEV\_T}(6, x)}{\sqrt{(1 - x^2)}}\,dx = 0$$

#4: $$\int_{-1}^{1} \frac{\text{CHEBYCHEV\_T}(3, x)^2}{\sqrt{(1 - x^2)}}\,dx = \frac{\pi}{2}$$

#5: $$\int_{-1}^{1} \frac{\text{CHEBYCHEV\_T}(6, x)^2}{\sqrt{(1 - x^2)}}\,dx = \frac{\pi}{2}$$

Hermite Polynomials are defined in $(-\infty, +\infty)$ and have the weight function $\omega(x) = e^{-x^2}$. Their definition is:

$$\text{HERMITE\_H}(n, x) := (-1)^n \cdot e^{x^2} \cdot \left(\frac{d}{dx}\right)^n e^{-x^2}$$

$$\int_{-\infty}^{\infty} H_n(x)H_m(x)e^{-x^2}\,\mathrm{d}x = \begin{cases} 0 & \text{for } n \neq m \\ 2^n n!\sqrt{\pi} & \text{for } n = m \end{cases}$$

```
#10:  herm(n) := ELEMENT(ITERATE([ELEMENT(w, 2), 2·x·ELEMENT(w, 2) - 2·(ELEMENT(w, 3) +
        1)·ELEMENT(w, 1), ELEMENT(w, 3) + 1], w, [1, 2·x, 0], n), 1)

#11:  VECTOR(herm(k), k, 0, 10)
```

#12: $\left[1,\ 2\cdot x,\ 4\cdot x^2 - 2,\ 4\cdot x\cdot(2\cdot x^2 - 3),\ 4\cdot(4\cdot x^4 - 12\cdot x^2 + 3),\ 8\cdot x\cdot(4\cdot x^4 - 20\cdot x^2 + 15),\ 8\cdot(8\cdot x^6 - 60\cdot x^4 + 90\cdot x^2 - 15),\ 16\cdot x\cdot(8\cdot x^6 - 84\cdot x^4 + 210\cdot x^2 - 105),\ 16\cdot(16\cdot x^8 - 224\cdot x^6 + 840\cdot x^4 - 840\cdot x^2 + 105),\ 32\cdot x\cdot(16\cdot x^8 - 288\cdot x^6 + 1512\cdot x^4 - 2520\cdot x^2 + 945),\ 32\cdot(32\cdot x^{10} - 720\cdot x^8 + 5040\cdot x^6 - 12600\cdot x^4 + 9450\cdot x^2 - 945)\right]$

#18: VECTOR(herm(k), k, 0, 10) - VECTOR(HERMITE_H(k), k, 0, 10)

#19: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

#20: $\int_{-\infty}^{\infty} e^{-x^2} \cdot \text{herm}(4) \cdot \text{herm}(6)\, dx = 0$

#21: $\int_{-\infty}^{\infty} e^{-x^2} \cdot \text{herm}(13) \cdot \text{herm}(27)\, dx = 0$

#22: $\int_{-\infty}^{\infty} e^{-x^2} \cdot \text{herm}(4) \cdot \text{herm}(4)\, dx = 384 \cdot \sqrt{\pi}$

#23: $\text{VECTOR}\left(\int_{-\infty}^{\infty} e^{-x^2} \cdot \text{herm}(k) \cdot \text{herm}(k)\, dx, k, 0, 5\right)$

#24: $[\sqrt{\pi}, 2\cdot\sqrt{\pi}, 8\cdot\sqrt{\pi}, 48\cdot\sqrt{\pi}, 384\cdot\sqrt{\pi}, 3840\cdot\sqrt{\pi}]$

#25: $\text{VECTOR}(2^k \cdot k! \cdot \sqrt{\pi}, k, 0, 5) = [\sqrt{\pi}, 2\cdot\sqrt{\pi}, 8\cdot\sqrt{\pi}, 48\cdot\sqrt{\pi}, 384\cdot\sqrt{\pi}, 3840\cdot\sqrt{\pi}]$

It could be a nice challenge for the students to find out the general rule to generate sequence #24

#26: $\int_{-\infty}^{\infty} e^{-x^2} \cdot \text{herm}(15) \cdot \text{herm}(15)\, dx = 42849873690624000 \cdot \sqrt{\pi}$

#27: $2^{15} \cdot 15! \cdot \sqrt{\pi} = 42849873690624000 \cdot \sqrt{\pi}$

In addition to the recurrence equation given in Marko Horbatsch´s contribution the *Hermite Polynomials* follow another recurrence equation which is connected to a differential equation:

$$H_n''(x) - 2xH_n'(x) + 2n\,H(x) = 0$$

$$\text{VECTOR}\left(\left(\frac{d}{dx}\right)^2 \text{herm}(k) - 2\cdot x \cdot \frac{d}{dx}\,\text{herm}(k) + 2\cdot k\cdot \text{herm}(k), k, 10, 15\right)$$

[0, 0, 0, 0, 0, 0]

$$H_n'(x) = 2n\,H_{n-1}(x)$$

$$\text{VECTOR}\left(\frac{d}{dx}\,\text{herm}(k), k, 10, 20\right) - \text{VECTOR}(2\cdot k\cdot \text{herm}(k-1), k, 10, 20)$$

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

Hermite Polynomials are also defined using the weight function $\omega(x) = e^{-\frac{x^2}{2}}$.

Last July I received a letter from Finland

"What do you like my files? The research in mathematics and physics (mechanics) has been long time my work, although I am not a professional physicist. I am a writer (middle level studies in physics, too), and I have new ideas about mechanics. Are the files suitable for circulation? The mathematics in these files is very simple."

I'm not a physicist, too, so I can't give a comment on Mr Ahonen's ideas. The time he has spent is as remarkable (a lot of hours for calcultating and plotting) as his enthusiasm. When I wrote to Finland, that I would like to publish his contribution I received 5 versions of Ahonen's Mechanics, one time two letters a day. In December when I informed Mr Ahonen that his ideas would be published in March 93 he seemed to be a bit disappointed: "So much work and now so long to wait! This is for me the most important thing, and for the world, too".

But finally, here is

The Mechanics of Erkki Ahonen (Part 1)

The Mechanics of Erkki Ahonen
Main Features
Erkki Ahonen Salo Finland
[DERIVE V.1.22 used. V.2.51 does not work well]

I have been working at the basic mechanics many years and have got interesting results. The fundamental result of my research is the equation of motion

$$a - a_0\left(\frac{u}{u_0}\right)^p \times \sqrt{\left(d^2 \pm \left(\frac{u}{u_0}\right)^k\right)^n} = 0, \qquad (1)$$

where $a = d^2x/dt^2 = du/dt = u\ du/dx$ = acceleration,
a_0 = constant acceleration;
$u = dx/dt$ = velocity, u_0 = constant velocity; $p,k,n \in R$
d = real or imaginary number.
x = coordinate of place, t = coordinate of time.

The integration constants x_0,t_0 move the origin.

EXAMPLE: The equation of the parabola is

$$a_0(t-t_0)=\pm\sqrt{2a_0(x-x_0)}\,,$$

when p,n = 0 in Eq. (1): $a = a_0$.

Eq.(1) gives as solutions a large set of curves when the parameters are changed.

DYNAMICS: The equation of force is

$$f(t)=m_0\frac{d}{dt}\frac{u}{bL}, \tag{2}$$

where m_0 = rest mass; $bL=\sqrt{1-\left(\frac{u}{c}\right)^2}$;;

c = velocity of light.

With the chain rule $\left(u\frac{dt}{dx}=1\right)$)

$$f(x)=m_0\,u\frac{d}{dx}\left(\frac{u}{bL}\right). \tag{3}$$

The total energy of a body is

$$E=\frac{m_o c^2}{bL}. \tag{4}$$

Derivating Eq. (4) one gets

$$\frac{dE}{dx}=m_0c^2\frac{d}{dx}\left(\frac{1}{bL}\right). \tag{5}$$

Let the right sides of Eq.(5) and Eq.(3) be equal,

$$m_0c^2\frac{d}{dx}\left(\frac{1}{bL}\right)=m_0\,u\frac{d}{dx}\left(\frac{u}{bL}\right). \tag{6}$$

Multiplying both sides of Eq.(6) by the factor dx,

$$m_0c^2\,d\left(\frac{1}{bL}\right)=m_0\,u\,d\left(\frac{u}{bL}\right). \tag{7}$$

Integrating of Eq. (7),

$$m_0c^2\int d\frac{1}{bL}=m_0\int u\,d\left(\frac{u}{bL}\right). \tag{8}$$

The left hand side of Eq. (8) is the same as the total energy E. The integration by parts in the right hand side.

$$\begin{aligned} E &= m_0 \int u\, d\left(\frac{u}{bL}\right) = m_0\left(\frac{u^2}{bL} + c^2 \int -\left(\frac{u}{c^2}\right)\frac{du}{bL}\right) = \\ &= m_0\left(\frac{u^2}{bL} + c^2\frac{bL^2}{bL}\right) = \qquad (9) \\ &= m_0\left(\frac{u^2}{bL} + \frac{c^2}{bL} - \frac{u^2}{bL}\right) = \frac{m_0 c^2}{bL}, \text{ or} \end{aligned}$$

$$f(x) = \frac{dE}{dx}, \qquad (10)$$

when the left hand sides on Eq.(3) and Eq.(5) are equal. Simplifying the right sides of Eq.(2),(3),(5) one gets

$$f = m_0 \frac{a}{bL^3}\ . \qquad (11)$$

Power = P,

$$\begin{aligned} P &= \frac{dE}{dt} \\ &= m_0 c^2 \frac{d}{dt}\left(\frac{1}{bL}\right) \qquad (12) \\ &= m_0 c^2 \frac{u}{c^2}\frac{du}{dt}\frac{1}{bL^3} \\ &= f \cdot u \end{aligned}$$

In general the force is

$$F(t) = m_0 \frac{d}{dt}\left(\frac{U}{bL}\right), \qquad (13)$$

where $bL = \sqrt{1 - U \cdot \frac{U}{c^2}}$. (The capital letters R,U,A,F are vectors.)

$$\begin{aligned} F &= m_0 \frac{d}{dt}\left(\frac{U}{bL}\right) \\ &= m_0\left(\frac{dU}{dt} bL^2 \frac{1}{bL^3} + U \cdot \frac{U}{c^2}\frac{dU}{dt}\right)\frac{1}{bL^3} \qquad (14) \\ &= m_0\left(\frac{dU}{dt} - U \cdot \frac{U}{c^2}\frac{dU}{dt} + U \cdot \frac{U}{c^2}\frac{dU}{dt}\right)\frac{1}{bL^3} \\ &= m_0 \frac{A}{bL^3} \end{aligned}$$

$$
\begin{aligned}
P &= \frac{dE}{dt} \\
&= m_0 c^2 \frac{d}{dt}\frac{1}{bL} \\
&= m_0 c^2 \left(\frac{U}{c^2}\right)\cdot\left(\frac{dU}{dt}\right)\frac{1}{bL^3} \qquad (15) \\
&= m_0 \left(\frac{dU}{dt}\right)\frac{1}{bL^3}\cdot U \\
&= F\cdot U
\end{aligned}
$$

or one can write

$$F\cdot U = \frac{dE}{dt}. \qquad (16)$$

From (16) one gets $\left(\frac{1}{U} = \frac{dt}{dR}\right)$

$$
\begin{aligned}
F &= \frac{dE}{dR} \\
&= m_0 c^2 \frac{d}{dR}\left(\frac{1}{bL}\right) \\
&= m_0 c^2 \left(\frac{U}{c^2}\right)\cdot\left(\frac{dU}{dR}\right)\frac{1}{bL^3} \qquad (17) \\
&= m_0 \left(ux\,I\cdot\frac{dux\_I}{dR} + uy\,J\cdot\frac{duy\,J}{dR}\right)\frac{1}{bL^3},
\end{aligned}
$$

ux I, uy J are the component vectors of vector U, and are functions of x, y. I, J are the unit vectors.

Remembering dR = dx I + dy J, one gets

$$
\begin{aligned}
F &= \left(\frac{ux\,I\,dux\,I\cdot I}{I}\cdot\left(dx\,I + dy\,J\right) + \frac{uy\,J\,duy\,J\cdot J}{J}\cdot\left(dx\,I + dy\,J\right)\right)\frac{1}{bL^3} \\
&= m_0\left(ux\,I\frac{dux}{dx} + uy\,J\frac{duy}{dy}\right)\frac{1}{bL^3} \\
&= m_0\left(ax\,I + ay\,J\right)\frac{1}{bL^3} \\
&= m_0\,A\frac{1}{bL^3},
\end{aligned}
$$

or

$$F(R) = \frac{dE}{dR} \qquad (18)$$

$$= \text{grad } E. \qquad (19)$$

$$F = m_0 \frac{A}{bL^3} \qquad (20)$$

$$A = \frac{dU}{dt} = U \cdot \frac{dU}{dR}. \qquad (21)$$

The upgrowth of the kinetic energy in $[R_1, R_2]$:

$$\int_{R_1}^{R_2} F \cdot dR = m_0 c^2 \int_{u_1}^{u_2} \left(\frac{u}{c^2}\right) \frac{du}{bL^3} = m_0 c^2 \left. \frac{1}{bL} \right|_{u_1}^{u_2} = E_2 - E_1 \qquad (22)$$

when $U \cdot dU = u\,du$; $U \cdot U = u^2$.

In the last two files (The Quantization of Mathematics 1 and 2) is used the polar coordinate system r,θ , and its conformal mapping to the rectangular plane. The calculations typically go:

$$r,\theta:\ u = \frac{dr}{d\theta}:\ a = \frac{du}{d\theta}$$

$$a = -u^{-1}\sqrt{(d^2 - u^2)^{-7}} \qquad (23)$$

$$\int -u\sqrt{(d^2 - u^2)^7}\,du \qquad (24)$$

$$\frac{\sqrt{(d^2 - u^2)^9}}{9} = \theta \qquad (25)$$

$$r1 = \int \sqrt{d^2 - (9\theta)^{2/9}}\,d\theta \qquad (26)$$

$$r2 = \int -\sqrt{d^2 - (9\theta)^{2/9}}\,d\theta \qquad (27)$$

Simplify the right hand side of (26) giving (28) and simplify the right hand side of (27) giving (29).

(28),(29) when substituted d = 5 can be plotted separately in some interval and in some scale, and they can be summed: ((28) + (29)) and then plotted. The sum of (28),(29) is analytically = 0, but with DERIVE's computing technique the program plots very interesting results. DERIVE's version 2.51 does not do the same.

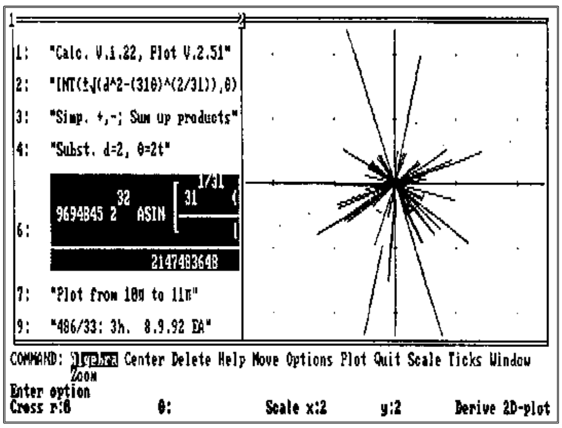
The calculations, and especially plottings, take long time; I have plotted with my 486/33 two or three days.

Why does the program plot the sum of (28),(29)? I think it makes use of roundings in calculations, which causes quantum effects in plotting. (28) and (29) are not identically the same, with reverse sign, and roundings are different in both. But together they form logical pictures in plottings, which I think the nature uses in the so called quantum level.

[Erkki Ahonen: Kappaleen liikkeen kuvaaminen x,t-tasolla, Halikko 1983, ISBN 951-99465-7-8]

The first two examples: Straight Line and Parabola

```
1:  "THE STRAIGHT LINE"
2:  "a-a0(u/u0)^p√(d^2±(u/u0)^k)^n=0"
3:  "a=d^2x/dt^2=du/dt=udu/dx:u=dx/dt"
4:  "a0=0;a=0;u=u0"
5:  "u0=Integration constant"
6:  "u0t=x-x0"
7:  "x0=Integration constant"
8:  "u0=x0=0.5"
9:  1/2*t=x-1/2
10:  t=2*x-1
11:  "plot straight line #10"
12:  "u0t=x-x0"
13:  "u0 = -0.5; x0 = -0.5"
14:  -1/2*t=x+1/2
15:  t=-2*x-1
16:  "plot #16"
```




```
1:  "THE PARABOLA"
2:  "a-a0(u/u0)^p√(d^2±  (u/u0)^k)^n=0"
3:  "a=d^2x/dt^2 = du/dt = u du/dx: u =dx/dt"
4:  "p = n = 0; a = a0; a = du/dt"
5:  "du = a0 dt: u = a0 t"
6:  "dx = a0 t dt: a0 = 1"
7:  INT(t,t)
8:  t^2/2=x
9:  t=-SQRT(2)*SQRT(x)
10:  t=SQRT(2)*SQRT(x)
11:  "plot #9, #10"
12:  "p = n = 0; a = a0; a = du/dx"
13:  "u du = a0 dx: u^2/2 = a0 x : a0 = 1"
14:  u^2/2=x
15:  u=-SQRT(2)*SQRT(x)
16:  u=SQRT(2)*SQRT(x)
17:  "dx/±√(2x) = dt"
18:  INT(1/SQRT(2*x),x)
19:  SQRT(2*x)=t
20:  -SQRT(2*x)=t
21:  "The equation of the parabola:"
22:  "a0 (t - t0) = ±√(2 a0(x - x0))"
23: "a0 = 2;t0 = x0 = 1/2"
24:  2*(t-1/2)=SQRT(4*(x-1/2))
25:  t=(SQRT(2)*SQRT(2*x-1)+1)/2
26:  2*(t-1/2)=-SQRT(4*(x-1/2))
27:  t=(1-SQRT(2)*SQRT(2*x-1))/2
28:  "plot #25, #27"
```

Will be continued in the next issue!
See a plot of Erkki´s "Quantization".

Trigonometry for the Classroom

Josef Böhm, Würmla

Yes. I know that the next contribution is easily to be done with any programming language. But, when I'm working for school or for private purposes I don't want to look for a while in my toolbox to find the fitting instrument. So, when I'm working with DERIVE, why don't use DERIVE for calculating triangles, too.

On the other hand I think that this application can be easily done with students. I give the starting points and let them continue the work. We can train simple CASE-decisions, abstractions and at last the use of a selfmade tool.

```
(sides s1, s2, s3 with opposite angles a1, a2, a3 (in degrees)

#1:   Calculation of Triangles – Dreiecksberechnungen

#2:   the 3rd angle – der Ergänzungswinkel

#3:   R_A(a1, a2) := 180° – a1 – a2

#4:   Triangle's inequation – Dreiecksungleichung

      TR_INEQU(s1, s2, s3) :=
        If s1 + s2 < s3 ∨ s2 + s3 < s1 ∨ s1 + s3 < s2
#5:        1
           0

#6:   The Cosine Rule – Der Kosinussatz
```

#7: $\text{CS\_OPPA3}(s1, s2, a3) := \sqrt{(s1^2 + s2^2 - 2\cdot s1\cdot s2\cdot \text{COS}(a3))}$

#8: $\text{CA\_OPPS1}(s1, s2, s3) := \dfrac{\text{ACOS}\left(\dfrac{s2^2 + s3^2 - s1^2}{2\cdot s2\cdot s3}\right)}{1°}$

```
#9:   The Sine Rule – Der Sinussatz
```

#10: $\text{SS\_OPPA2}(s1, a1, a2) := \dfrac{s1\cdot \text{SIN}(a2)}{\text{SIN}(a1)}$

#11: $\text{SA\_OPPS2}(s1, s2, a1, k) := \dfrac{\text{ASIN}\left(\dfrac{s2\cdot \text{SIN}(a1)}{s1}\right)}{1°}\cdot(1 - 2\cdot k) + 180\cdot k$

```
#12:  the sides s1,s2,s3 are known – die drei Seiten s1,s2,s3 sind gegeben:

      SSS(s1, s2, s3) :=
        If TR_INEQU(s1, s2, s3) = 1
#13:       "no triangle !"
           ["Angle opposite S1: ",CA_OPPS1(s1, s2, s3);"Angle opposite S2: ",



        CA_OPPS1(s2, s3, s1); "Angle opposite S3: ", CA_OPPS1(s3, s2, s1)]

#14:  The sides s1, s2 and the angle a3 between them are known:

#15:  zwei Seiten S1, S2 und der eingeschlossene Winkel A3 sind gegeben:

                             ⎡   Third side:               CS_OPPA3(s1, s2, a3)
#16:  SAS(s1,a3,s2) :=       ⎢ Angle opposite S1:   CA_OPPS1(s1, s2, CS_OPPA3(s1, s2,
                             ⎣ Angle opposite S2:   CA_OPPS1(s2, s1, CS_OPPA3(s1, s2,

              ⎤
        a3))  ⎥
        a3))  ⎦

#17:  The side s3 and the adjacent angles a1, a3 are known:

#18:  eine Seite S3 und die beiden anliegenden Winkel A1, A2 sind gegeben:

                               ⎡                          R_A(a1, a2)       ⎤
                               ⎢   Third angle:          ─────────────      ⎥
                               ⎢                              1°            ⎥
#19:  ASA(a1, s3, a2) :=       ⎢ Side opposite A1:   SS_OPPA2(s3, R_A(a1, a2), a1) ⎥
                               ⎣ Side opposite A2:   SS_OPPA2(s3, R_A(a1, a2), a2) ⎦

#20:  zwei Seiten S1 und S2 und der gegenüberliegende Winkel A1 sind gegeben:

#21:  hier können eine, keine oder zwei Lösungen auftreten!

#22:  two sides s1 and s2 and one opposite angle a1 are known:

#23:  one, none or two solutions!

#24:  zuerst eine Hilfsfunktion SSAH, dann die Fallunterscheidung: arbeite mit
        SSA(s1,s2,a1):

#25:  at first an auxiliary function SSAH, but work with SSA(...), CASE–
        decision:
```

```
                                        ┌ Solution: Triangle
                                        │
                                        │ Angle opposite S2:
                                        │
#26:  SSAH(s1, s2, a1, k) :=            │
                                        │    Third angle:
                                        │
                                        │
                                        └    Third side:
                                   k + 1                              ┐
                                                                      │
                          SA_OPPS2(s1, s2, a1, k)                     │
                                                                      │
                      R_A(a1, SA_OPPS2(s1, s2, a1, k)°)               │
                      ─────────────────────────────────               │
                                     1°                               │
                                                                      │
        SS_OPPA2(s1, a1, R_A(a1, SA_OPPS2(s1, s2, a1, k)°))           ┘

      SSA(s1, s2, a1) :=
        If s1 ≥ s2
           SSAH(s1, s2, a1, 0)
#27:       If s2·SIN(a1)/s1 > 1
              "no triangle !"
              VECTOR(SSAH(s1, s2, a1, k), k, 0, 1)

#28:  Some examples:

#29:  a = 102, b = 61, c = 109

#30:  SSS(102, 61, 109)

      ┌ Angle opposite S1:   66.99036 ┐
#31:  │ Angle opposite S2:   33.39848 │
      └ Angle opposite S3:   79.61111 ┘

#32:  a=100, b = 160, α = 33°

#33:  SSA(100, 160, 33°)

      ┌┌ Solution: Triangle       1     ┐ ┌ Solution: Triangle       2     ┐┐
      ││ Angle opposite S2:   60.62436  │ │ Angle opposite S2:   119.3756  ││
#34:  ││    Third angle:      86.37563  │,│    Third angle:      27.62436  ││
      └└    Third side:       183.2406  ┘ └    Third side:       85.13396  ┘┘

#35:  b = 85.3, c = 23.4, α = 58.6°

#36:  SAS(85.3, 58.6°, 23.4)

      ┌    Third side:       75.78758 ┐
#37:  │ Angle opposite S1:   106.1197 │
      └ Angle opposite S2:   15.28023 ┘
```

#38: a=100, b = 160, α = 53°

#39: SSA(100, 160, 53°)

#40: no triangle !

#41: a=160, b=100, α = 33°

#42: SSA(160, 100, 33°)

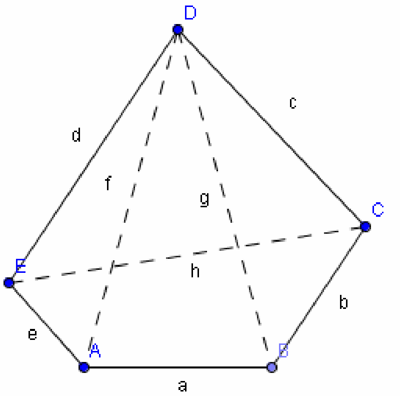
#43:
$$\begin{bmatrix} \text{Solution: Triangle} & 1 \\ \text{Angle opposite S2:} & 19.9012 \\ \text{Third angle:} & 127.0987 \\ \text{Third side:} & 234.312 \end{bmatrix}$$

Pentagon:

AB = 30, BC = 52, AE = 30,

AD = 65, BD = 60, EC = 68, ∠EAB = 106,1°

Find ED = d = ? and CD = c = ?



#44: Calculation of the pentagon – Fünfecksberechnung

#45: [AB := 30, BC := 52, AE := 30, AD := 65, BD := 60, EC := 68, EAB := 106.1°]

#46: EB = ?, AEB = ?

#47: SAS(AE, EAB, AB)

#48:
$$\begin{bmatrix} \text{Third side:} & 47.94961 \\ \text{Angle opposite S1:} & 36.95 \\ \text{Angle opposite S2:} & 36.95 \end{bmatrix}$$

#49: [EB := 47.9496, AEB := 36.95°]

#50: BEC = ?

#51: SSS(BC, EB, EC)

#52:
$$\begin{bmatrix} \text{Angle opposite S1:} & 49.68443 \\ \text{Angle opposite S2:} & 44.67621 \\ \text{Angle opposite S3:} & 85.63935 \end{bmatrix}$$

#53: BEC := 49.6844°

#54: DAB = ?, EAD = ?

#55: SSS(BD, AB, AD)

#56:
$$\begin{bmatrix} \text{Angle opposite S1:} & 66.98167 \\ \text{Angle opposite S2:} & 27.39935 \\ \text{Angle opposite S3:} & 85.61897 \end{bmatrix}$$

#57: EAD := EAB − 66.9816°

#58: ED = d = ?

#59: SAS(AE, EAD, AD)

#60:
$$\begin{bmatrix} \text{Third side:} & 45.81712 \\ \text{Angle opposite S1:} & 24.40063 \\ \text{Angle opposite S2:} & 116.4809 \end{bmatrix}$$

#61: d = 45.8171

#62: DEC := 116.4809° − AEB − BEC

#63: SAS(45.8171, DEC, EC)

#64:
$$\begin{bmatrix} \text{Third side:} & 36.31207 \\ \text{Angle opposite S1:} & 38.89904 \\ \text{Angle opposite S2:} & 111.2544 \end{bmatrix}$$

#65: c = 36.3121

I find it very useful to doublecheck the results by means of a plot in the 2D-plot window reproducing the sketch procedure how if it were done by hand or by using Dynamic Procedure. There are a lot of additonal advantages by transfering the abstract calculation to working with geometric objects and by connecting trigonometry with analytic geometry. I show the check performed with DERIVE and with my favourite Dynamic Geometry program GEOGEBRA which can be downloaded for free (for many languages) from www.geogebra.at. (2006)

#1: CaseMode := Sensitive

#2: InputMode := Word

#3: [A := [0, 0], B := [30, 0]]

#4: $k1 := x^2 + y^2 = 65^2$

#5: $\left[k1 := x^2 + y^2 = 65^2,\ k2 := (x - 30)^2 + y^2 = 60^2\right]$

#6: $\text{SOLUTIONS}(k1 \wedge k2, [x, y]) = \begin{bmatrix} \frac{305}{12} & \frac{5\cdot\sqrt{20615}}{12} \\ \frac{305}{12} & -\frac{5\cdot\sqrt{20615}}{12} \end{bmatrix}$

#7: $D := \left[\frac{305}{12}, \frac{5\cdot\sqrt{20615}}{12}\right]$

#8: $\left[k3 := x^2 + y^2 = 30^2,\ h1 := y = \text{TAN}(106.1°)\cdot x\right]$

#9: SOLUTIONS(k3 ∧ h1, [x, y])

#10: $\begin{bmatrix} 8.319439599 & -28.82337462 \\ -8.319439599 & 28.82337462 \end{bmatrix}$

#11: E := [-8.319439599, 28.82337462]

#12: $\left[k4 := (x - E_1)^2 + (y - E_2)^2 = 68^2,\ k5 := (x - 30)^2 + y^2 = 52^2\right]$

#13: SOLUTIONS(k4 ∧ k5, [x, y])

#14: $\begin{bmatrix} 58.00792189 & 43.81274142 \\ -4.32736149 & -39.05934271 \end{bmatrix}$

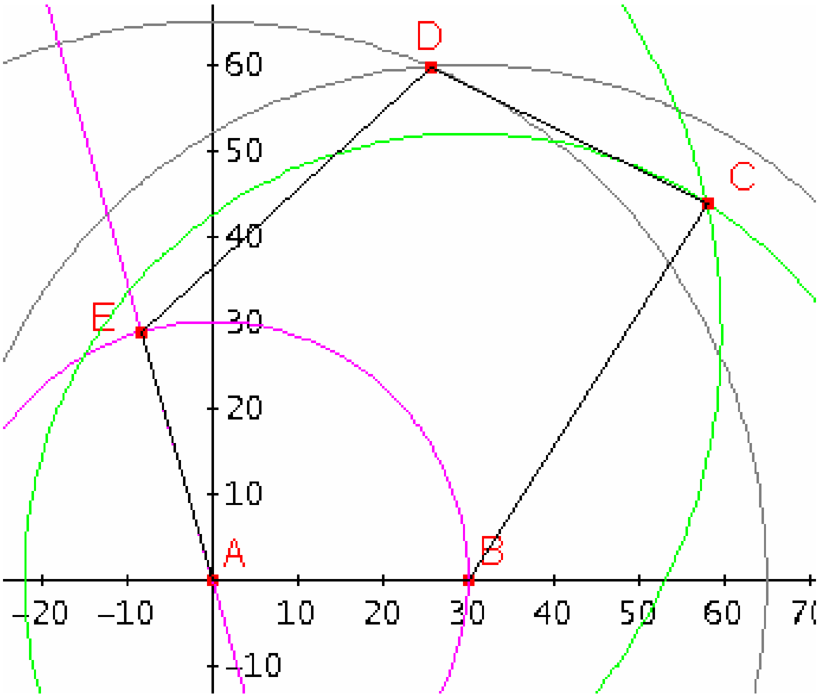
#15: C := [58.00792189, 43.81274142]

#16: [A, B, C, D, E, A]

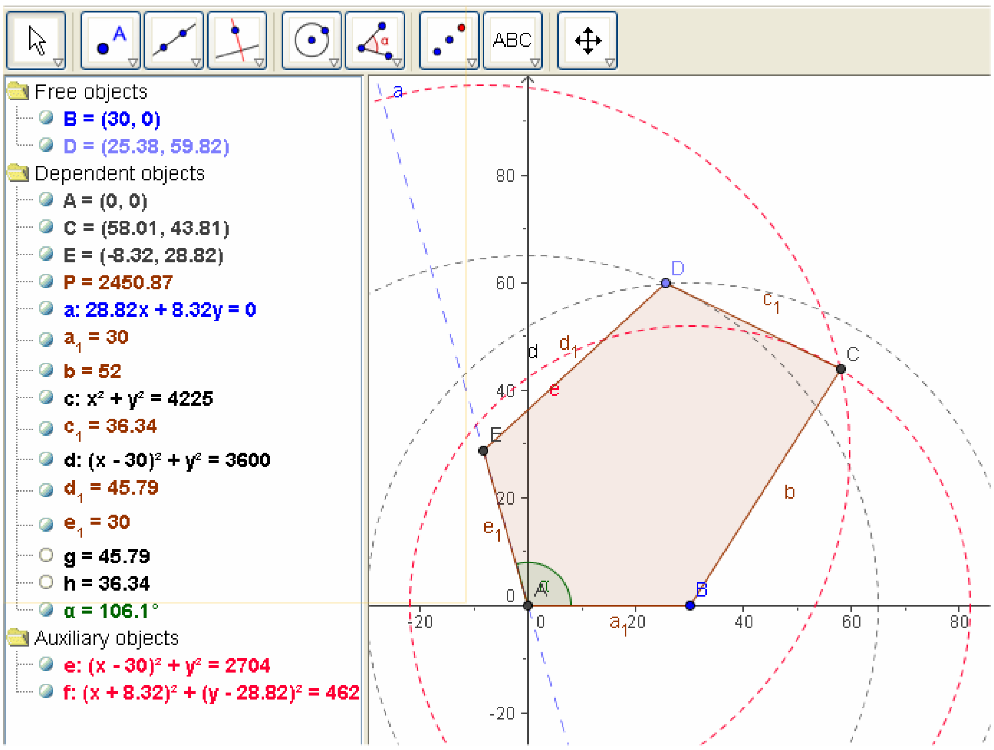
#17: $[|E - D|, |C - D|] = [45.81709477, 36.31215067]$

See the simultaneous DERIVE plot on the next page!

The DERIVE plot:



The English Geogebra construction (c_1 and d_1 are the measured lengths):



Extended *Fourier* Series in *DERIVE*

De Siquera & Domingos, Petropolis, Brasil

1 - ABSTRACT

Nowadays, the computers are very important to the progress of science and technology. Very complex algebraic calculations can be done by programs. We illustrate it with High Energy Physics and Engineering calculations. The algebraic calculations have showed to be a powerful tool for teaching Mathematics in the graduation courses also. Exhaustive calculations needed in the series expansion can be done quickly using algebraic computation. In this work we used DERIVE to implement the Fourier series with three functions in the interval. However, we cannot let the physical sense to escape from the students. In order to achieve this we solve a Strength of Materials Problem.

2 - INTRODUCTION

This work has an objective implement and extend an utility in the DERIVE program and also solves a problem in engineering Strength of Materials. This utility file implements the Fourier series until three functions in the interval. Nowadays, exists the Fourier series with just one function in the interval, in the utility file INT_APPS.MTH.

Some series has been successfully tested and this stimulates the students of the Engineering course to solve more elaborated problems.

3 - THE FOURIER SERIES

Consider a periodic function $y(x)$ in the interval $[a, a+2p]$ as showed in figure (1).

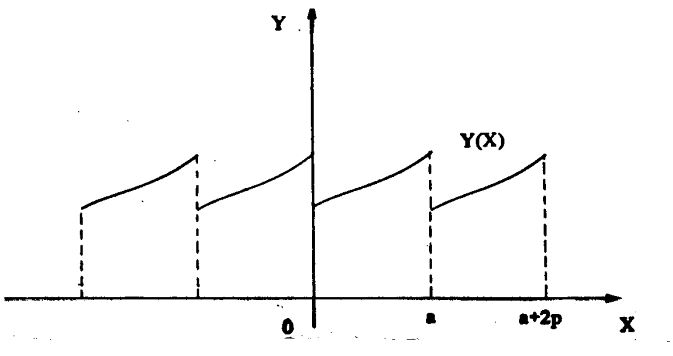


Figure 1. A periodic function $y(x)$

The period of the function in figure (1) is $2p$. This periodic function could be written as terms of sines and cosines as being (see [1])

$$y(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos\frac{n\pi x}{p} + \sum_{n=1}^{\infty} b_n \sin\frac{n\pi x}{p} \qquad (1)$$

were a_o, a_n and b_n are the coefficients of Euler or Euler-Fourier formulas and are:

$$a_0 = \frac{1}{p} \int_a^{a+2p} y(x)\,dx \tag{2}$$

$$a_n = \frac{1}{p} \int_a^{a+2p} y(x) \cos\frac{n\pi x}{p}\,dx \tag{3}$$

$$b_n = \frac{1}{p} \int_a^{a+2p} y(x) \sin\frac{n\pi x}{p}\,dx \tag{4}$$

where p is the semi-period.

Substituting the equations (2), (3) and (4) in (1), we would obtain the Fourier series.

There exists one theorem [2] that would not be showed here, that affirms, that for developing the Fourier series, the equations (2), (3) and (4) can be splitted in intervals $[x_1,x_2]$, $[x_2,x_3]$, ..., $[x_n,x_{n+1}]$, where each interval counts each function.

In this work we will limit ourselves to just three functions in the interval $[a,d]$ to development of the Fourier series. These generalized three functions are showed in figure (2)

The semi-period will be: $p = \dfrac{d-a}{2}$. (5)

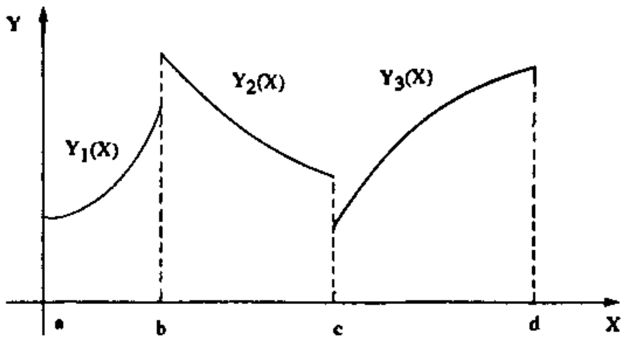


Figure 2. A periodic function with three functions in the interval [a,d]

Then the equations (2), (3) and (4) will be reduced to:

$$a_0 = \frac{1}{p}\left(\int_a^b y_1(x)\,dx + \int_b^c y_2(x)\,dx + \int_c^d y_3(x)\,dx \right) \tag{6}$$

$$a_n = \frac{1}{p}\left(\int_a^b y_1(x)\cos\frac{n\pi x}{p}\,dx + \int_b^c y_2(x)\cos\frac{n\pi x}{p}\,dx + \int_c^d y_3(x)\cos\frac{n\pi x}{p}\,dx \right) \tag{7}$$

$$b_n = \frac{1}{p}\left(\int_a^b y_1(x)\sin\frac{n\pi x}{p}\,dx + \int_b^c y_2(x)\sin\frac{n\pi x}{p}\,dx + \int_c^d y_3(x)\sin\frac{n\pi x}{p}\,dx \right) \tag{8}$$

with p showed in equation (5).

Substituting the equations (6), (7) and (8) in (1) will result in Fourier series of the functions showed in figure (2).

4 - THE IMPLEMENTATION IN DERIVE

For the implementation in DERIVE we define two functions, as showed below:

`FOURIER_2FUNCTION(x,a,b,c,n,y1,y2) = RHS Equation(1)` (9)

`FOURIER_3FUNCTION(x,a,b,c,d,n,y1,y2,y3) = RHS Equation(2)` (10)

where:

`FOURIER_NFUNCTION( )` – Name of the function defined in DERIVE;

- x - Variable that we intend to integrate;
- a, b, c, d - Borders of the intervals;
- n - Number of desired terms;
- $y1$, $y2$, $y3$ - Functions in interval.

By this way the work of expanding will be extremely reduced.

5 - APPLICATION

One application that grips engineering students was the fast solution of a complex problem of Strength of Materials, using the implementation in this work.

Consider the beam support the load showed in figure (3)

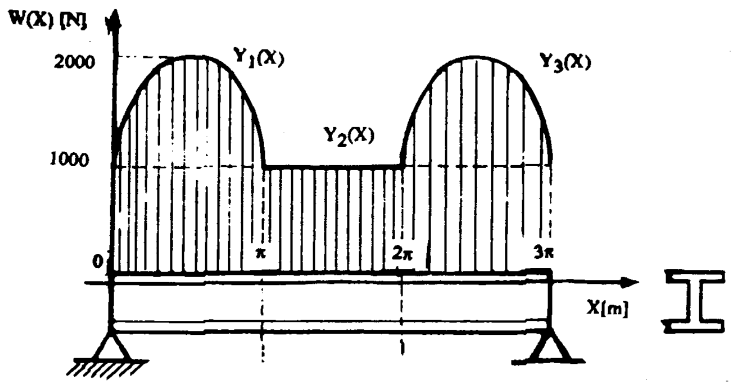


Figure 3. The beam to solution of engineering problem

In the interval [0,π] the distribution of the loading is sinoidal, between π and 2π it is uniformly distributed and in [2π,3π] the loading is sinoidal again. Then for figure (3) we will write:

$$y(x)=\begin{cases}1000\,(1+\sin x)\ \mathrm{Nm}^{-1} & \text{for } 0\le x\le \pi\\ 1000\ \mathrm{Nm}^{-1} & \text{for } \pi\le x\le 2\pi\\ 1000\,(1+\sin x)\ \mathrm{Nm}^{-1} & \text{for } 2\pi\le x\le 3\pi\end{cases} \tag{11}$$

Only one function could be used for describing the loading. Then, using the Fourier series and the implementation made in this work we can write:

`FOURIER_3FUNCTION(x,0,π,2π,3π,n,1000(1+sin(x),1000,1000(1+sin(x))`

Figure (4) shows the developing of the Fourier series with $n = 1$ and $n = 6$.

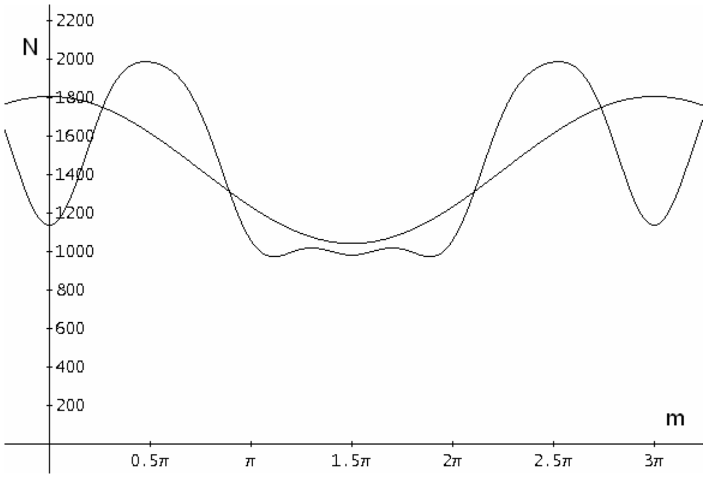


Figure 4. The loading developed in Fourier series

The problem of Strength of Materials resumes in determining the bending moment diagram and the deflection of the beam. The solution for this problem would consist in solving the differential equation [3].

$$E\,I\,\frac{d^4y}{dx^4} = -W(x) \qquad (12)$$

with E = Modulus of Elasticity;

I = Moment of Inertia;

$W(x)$ = Loading.

Equation (12) integrated two times gives us the bending moment:

$$E\,I\,\frac{d^2y}{dx^2} = -M(x) = -\int dx \int W(x)\,dx + c_1x + c_2 \qquad (13)$$

The boundaries conditions are: $M(x = 0) = 0$ and $M(x = 3\pi) = 0$.

Then we have c_1 and c_2. To solve equation (13) we define in DERIVE the function:

```
MOMENT(x,c1,c2,W(x)) = RHS Equation (13)
```
(14)

We will rapidly find c_1 and c_2 equalling the above result to zero.

The bending diagram is showed in figure (5).

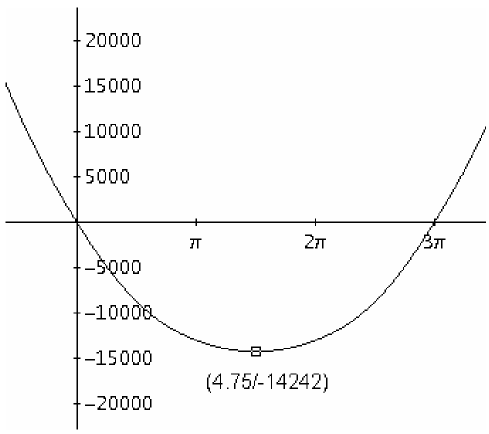


Figure 5. The bending moment diagram

The maximum can be found by the graphic and your value is M = 14242 Nm.

(You can also differentiate and find the extremal value.)

Integrating equation (13) two times we will find the function $y(x)$ or the deflection of the beam. For this purpose we will define a new function in DERIVE:

Integrating equation (13) two times we will find the function $y(x)$ or the deflection of the beam. For this purpose we will define a new function in DERIVE:

$$\text{FUNCTION}(x, c3, c4, m) := \frac{-\int dx\ (\int m\ dx + c3) + c4}{E \cdot I}$$

where c_3 and c_4 are constants that are fond by the boundaries conditions:

FUNCTION($x = 0$) = 0 and FUNCTION($x = 3\pi$) = 0.

(You will find the calculation at the end of the contribution, Editor)

The deflection diagram is showed in Figure 6 with E = 210 GPa and $I = 2.98 \times 10^4\ \text{m}^4$.

(H-Beam W 250 × 167)

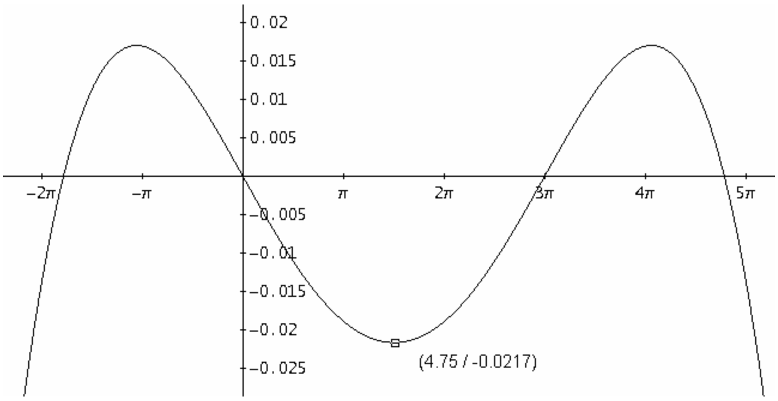


Figure 6. The deflection of the beam

The own weight of the beam was despized. We can find the maximum of y(x) by the graphic:

Y(x)max = 0.0217 m.

6 - CONCLUSION

The main objective of this work was to extend the utilities for the DERIVE program, the calculation of the Fourier series up until three functions. It has been also shown one application on Strength of Materials that would be almost impossible without the help of the DERIVE program.

Some problems of Heat Transfer, Electricita, Waves would also use the Fourier series and they could be easily solved with the implementation showed in this work.

7 - ACKNOWLEDGEMENT

The authors are grateful to the Universidfade Catolica de Petropolis for financial support.

8 - BIBLIOGRAPHY

[1] Boyce, William, Equacaes Diferenciaie Elementares e Problemas de Valores de Contorno, Ed Guanabara, RJ, 1968

[2] Willie, C.R., Advanced Engineering Mathematics, McGraw-Hill Book Company, 3o-edicao, 1960-1966

[3] Beer, Ferdinand P., Johnston, E., Russel jr, Resistencis dos Materials, McGraw-Hill do Brasil, 1987

Carlos Eduardo Reuther de Siqueira, Jorge Ricardo Valardan Domingos
Universidade Catolica de Petropolis, Inst. de Ciencias Exatas e Naturalis, Grupo de Fisica Teorica
Rua Barao do Amazonas, 124 - Centro, 25600 Petropolis - RJ - BRA

#1: $\text{FOURIER\_2FUNCTION}(x, a, b, c, n, y1, y2) := \frac{1}{c - a}\cdot\left(\int_a^b y1\,dx + \int_b^c y2\,dx\right) + \frac{2}{c - a}\cdot\sum_{k=1}^{n} \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot\left(\int_a^b \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot y1\,dx + \int_b^c \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot y2\,dx\right) + \frac{2}{c - a}\cdot\sum_{k=1}^{n} \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot\left(\int_a^b \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot y1\,dx + \int_b^c \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{c - a}\right)\cdot y2\,dx\right)$

#2: $\text{FOURIER\_3FUNCTION}(x, a, b, c, d, n, y1, y2, y3) := \frac{1}{d - a}\cdot\left(\int_a^b y1\,dx + \int_b^c y2\,dx + \int_c^d y3\,dx\right) + \frac{2}{d - a}\cdot\sum_{k=1}^{n} \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot\left(\int_a^b \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y1\,dx + \int_b^c \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y2\,dx + \int_c^d \text{COS}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y3\,dx\right) + \frac{2}{d - a}\cdot\sum_{k=1}^{n} \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot\left(\int_a^b \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y1\,dx + \int_b^c \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y2\,dx + \int_c^d \text{SIN}\left(\frac{2\cdot k\cdot \pi\cdot x}{d - a}\right)\cdot y3\,dx\right)$

#3: $\text{FOURIER\_3FUNCTION}(x, 0, \pi, 2\cdot\pi, 3\cdot\pi, 1, 1000\cdot(1 + \text{SIN}(x)), 1000, 1000\cdot(1 + \text{SIN}(x)))$

#4: $\frac{1200\cdot \text{COS}\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{1000\cdot(3\cdot\pi + 4)}{3\cdot\pi}$

#5: $\text{IF}\left(x \ge 0 \wedge x \le 3\cdot\pi, \frac{1200\cdot \text{COS}\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{1000\cdot(3\cdot\pi + 4)}{3\cdot\pi}\right)$

#6: FOURIER_3FUNCTION(x, 0, π, 2·π, 3·π, 2, 1000·(1 + SIN(x)), 1000, 1000·(1 + SIN(x)))

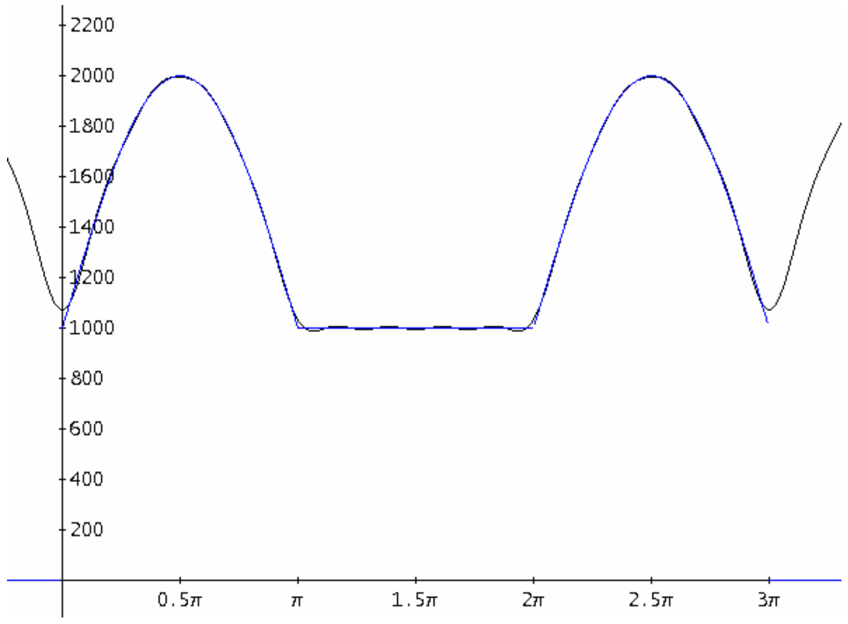
#7: $$- \frac{6000\cdot \cos\left(\frac{4\cdot x}{3}\right)}{7\cdot \pi} + \frac{1200\cdot \cos\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{1000\cdot (3\cdot \pi + 4)}{3\cdot \pi}$$

#8: FOURIER_3FUNCTION(x, 0, π, 2·π, 3·π, 6, 1000·(1 + SIN(x)), 1000, 1000·(1 + SIN(x)))

#9: $$- \frac{1600\cdot \cos(4\cdot x)}{9\cdot \pi} - \frac{6000\cdot \cos\left(\frac{10\cdot x}{3}\right)}{91\cdot \pi} - \frac{1200\cdot \cos\left(\frac{8\cdot x}{3}\right)}{11\cdot \pi} - \frac{8000\cdot \cos(2\cdot x)}{9\cdot \pi} - \frac{6000\cdot \cos\left(\frac{4\cdot x}{3}\right)}{7\cdot \pi} + \frac{1200\cdot \cos\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{1000\cdot (3\cdot \pi + 4)}{3\cdot \pi}$$

#10: $$\mathrm{IF}\left(x \ge 0 \wedge x \le 3\cdot\pi,\ - \frac{1600\cdot \cos(4\cdot x)}{9\cdot \pi} - \frac{6000\cdot \cos\left(\frac{10\cdot x}{3}\right)}{91\cdot \pi} - \frac{1200\cdot \cos\left(\frac{8\cdot x}{3}\right)}{11\cdot \pi} - \frac{8000\cdot \cos(2\cdot x)}{9\cdot \pi} - \frac{6000\cdot \cos\left(\frac{4\cdot x}{3}\right)}{7\cdot \pi} + \frac{1200\cdot \cos\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{1000\cdot (3\cdot \pi + 4)}{3\cdot \pi}\right)$$

#11: FOURIER_3FUNCTION(x, 0, π, 2·π, 3·π, 12, 1000·(1 + SIN(x)), 1000, 1000·(1 + SIN(x)))



Set w:=#8 and let´s proceed with this approximation for the given function

#12: w ≔ FOURIER_3FUNCTION(x, 0, π, 2·π, 3·π, 6, 1000·(1 + SIN(x)), 1000, 1000·(1 + SIN(x)))

#13: MOMENT(x, c1, c2, w) ≔ ∫ ∫ w dx dx + c1·x + c2

#14: $$\frac{100\cdot COS(4\cdot x)}{9\cdot\pi} + \frac{540\cdot COS\left(\frac{10\cdot x}{3}\right)}{91\cdot\pi} + \frac{675\cdot COS\left(\frac{8\cdot x}{3}\right)}{44\cdot\pi} + \frac{2000\cdot COS(2\cdot x)}{9\cdot\pi} + \frac{3375\cdot COS\left(\frac{4\cdot x}{3}\right)}{7\cdot\pi} - \frac{2700\cdot COS\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{3\cdot\pi\cdot(500\cdot x^2 + c1\cdot x + c2) + 2000\cdot x^2}{3\cdot\pi}$$

#15: x = 0, M = 0

#16: $$c2 - \frac{3368935}{1716\cdot\pi}$$

#17: $$c2 = \frac{3368935}{1716\cdot\pi}$$

#18: x = 3π, M = 0, c2 = 3368935 / (1716 π))

#19: 3·π·(1500·π + c1 + 2000)

#20: c1 = − 500·(3·π + 4)

#21: substitute for c1 and c2:

#22: $$\frac{100\cdot COS(4\cdot x)}{9\cdot\pi} + \frac{540\cdot COS\left(\frac{10\cdot x}{3}\right)}{91\cdot\pi} + \frac{675\cdot COS\left(\frac{8\cdot x}{3}\right)}{44\cdot\pi} + \frac{2000\cdot COS(2\cdot x)}{9\cdot\pi} + \frac{3375\cdot COS\left(\frac{4\cdot x}{3}\right)}{7\cdot\pi} - \frac{2700\cdot COS\left(\frac{2\cdot x}{3}\right)}{\pi} + \frac{3\cdot\pi\cdot\left(500\cdot x^2 + (-500\cdot(3\cdot\pi + 4))\cdot x + \frac{3368935}{1716\cdot\pi}\right) + 2000\cdot x^2}{3\cdot\pi}$$

Substitute for c_1 and c_2 to obtain $m(x)$-expression #22, then set `m:=#22` and plot the bending moment diagram (Figure 5). You can try to find the extremal value.

#23: m := #22

#24: $$NSOLUTIONS\left(\frac{d}{dx}\ m = 0,\ x\right)$$

#25: [4.712388980]

#26: SUBST(m, x, 4.71238898)

#27: $- 1.424206935\cdot 10^{4}$

#28: $$\text{FUNCTION}(x, c3, c4, m) := \frac{-\int\left(\int m\,dx + c3\right)dx + c4}{e \cdot i}$$

#29: $$\frac{25\cdot\text{COS}(4\cdot x)}{36\cdot\pi\cdot e\cdot i} + \frac{243\cdot\text{COS}\left(\frac{10\cdot x}{3}\right)}{455\cdot\pi\cdot e\cdot i} + \frac{6075\cdot\text{COS}\left(\frac{8\cdot x}{3}\right)}{2816\cdot\pi\cdot e\cdot i} + \frac{500\cdot\text{COS}(2\cdot x)}{9\cdot\pi\cdot e\cdot i} + \frac{30375\cdot\text{COS}\left(\frac{4\cdot x}{3}\right)}{112\cdot\pi\cdot e\cdot i} - \frac{6075\cdot\text{COS}\left(\frac{2\cdot x}{3}\right)}{\pi\cdot e\cdot i} + \frac{2574000\cdot\pi^2\cdot x^3 - 3432\cdot\pi\cdot(125\cdot x^4 - 1000\cdot x^3 + 3\cdot c3\cdot x - 3\cdot c4) - 5\cdot x^2\cdot(114400\cdot x^2 + 2021361)}{10296\cdot\pi\cdot e\cdot i}$$

#30: Substitute for e = 210, i = 2.98 10^4 and the boundaries conditions: (→Figure 6)

#31: $$\frac{c4}{6258000} - \frac{350512647}{381821440000\cdot\pi}$$

#32: $$c4 = \frac{1051537941}{183040\cdot\pi}$$

#33: $$\frac{\pi\cdot(1287000\cdot\pi^3 + 1716000\cdot\pi^2 - 1144\cdot c3 - 3368935)}{2386384000}$$

#34: $$c3 = \frac{5\cdot(257400\cdot\pi^3 + 343200\cdot\pi^2 - 673787)}{1144}$$

#35: $$\frac{\text{COS}(4\cdot x)}{9011520\cdot\pi} + \frac{81\cdot\text{COS}\left(\frac{10\cdot x}{3}\right)}{949130000\cdot\pi} + \frac{81\cdot\text{COS}\left(\frac{8\cdot x}{3}\right)}{234967040\cdot\pi} + \frac{\text{COS}(2\cdot x)}{112644\cdot\pi} + \frac{81\cdot\text{COS}\left(\frac{4\cdot x}{3}\right)}{1869056\cdot\pi} - \frac{81\cdot\text{COS}\left(\frac{2\cdot x}{3}\right)}{83440\cdot\pi} - \frac{1853280000\cdot\pi^4\cdot x + 2471040000\cdot\pi^3\cdot x - 411840000\cdot\pi^2\cdot x^3 + 2400\cdot\pi\cdot x\cdot(28600\cdot x^3 - 228800\cdot x^2 - 2021361) + 91520000\cdot x^4 + 1617088800\cdot x^2 - 9463841469}{10309178880000\cdot\pi}$$

#36: More examples:

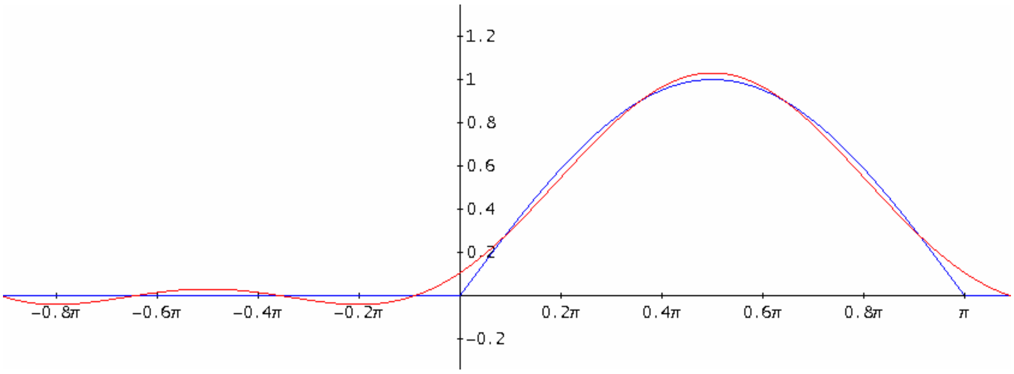
#37: Example 1: Two functions in the interval

38: Original function:

#39: [IF(x < −π ∨ x > π, 0), IF(x ≥ −π ∧ x ≤ 0, 0), IF(x ≥ 0 ∧ x ≤ π, SIN(x))]

#40: FOURIER_2FUNCTION(x, −π, 0, π, 3, 0, SIN(x))

#41: $$-\frac{2\cdot COS(2\cdot x)}{3\cdot \pi} + \frac{SIN(x)}{2} + \frac{1}{\pi}$$



#42: Example 2: Three functions in the interval

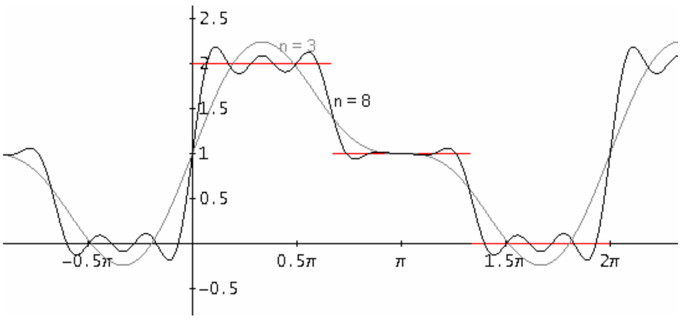
#43: $$\left[IF\left(x \ge 0 \wedge x \le \frac{2\cdot \pi}{3}, 2\right), IF\left(x \ge \frac{2\cdot \pi}{3} \wedge x \le \frac{4\cdot \pi}{3}, 1\right), IF\left(x \ge \frac{4\cdot \pi}{3} \wedge x \le 2\cdot \pi, 0\right)\right]$$

#44: $$FOURIER\_3FUNCTION\left(x, 0, \frac{2\cdot \pi}{3}, \frac{4\cdot \pi}{3}, 2\cdot \pi, 3, 2, 1, 0\right)$$

#45: $$\frac{3\cdot SIN(2\cdot x)}{2\cdot \pi} + \frac{3\cdot SIN(x)}{\pi} + 1$$

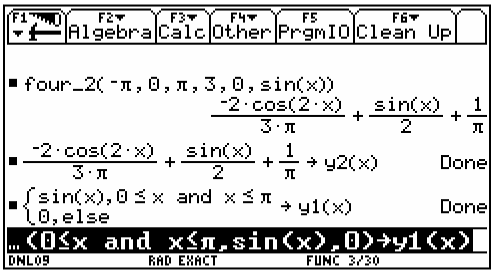
#46: $$FOURIER\_3FUNCTION\left(x, 0, \frac{2\cdot \pi}{3}, \frac{4\cdot \pi}{3}, 2\cdot \pi, 8, 2, 1, 0\right)$$

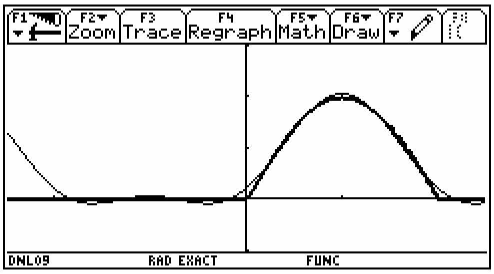
#47: $$\frac{3\cdot SIN(8\cdot x)}{8\cdot \pi} + \frac{3\cdot SIN(7\cdot x)}{7\cdot \pi} + \frac{3\cdot SIN(5\cdot x)}{5\cdot \pi} + \frac{3\cdot SIN(4\cdot x)}{4\cdot \pi} + \frac{3\cdot SIN(2\cdot x)}{2\cdot \pi} + \frac{3\cdot SIN(x)}{\pi} + 1$$

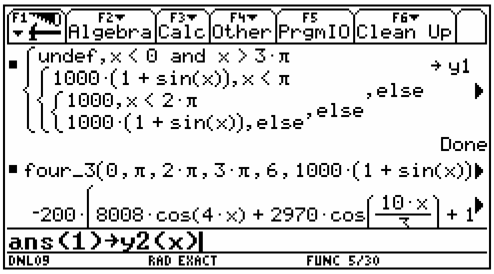


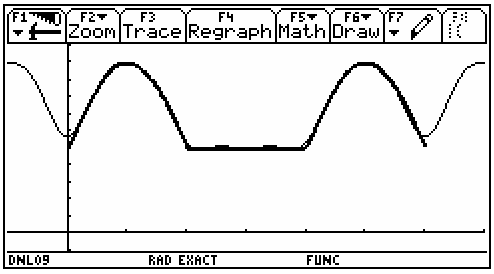
It is no problem to transfer the functions to the TI-device. Don´t include *x* into the variable list. It doesn´t work. I show the Fourier-series for two pieces within the period and perform the major part of the calculation of the beam. Josef

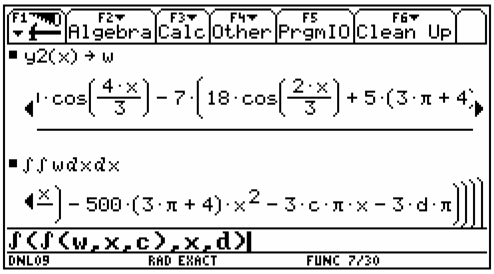
```
four_2(a,b,c,n,y1_,y2_)
Func
1/(c-a)*(∫(y1_,x,a,b)+∫(y2_,x,b,c))+2/(c-a)*Σ(cos(2*k*π*x/(c-a))*
(∫(cos(2*k*π*x/(c-a))*y1_,x,a,b)+∫(cos(2*k*π*x/(c-a))*
y2_,x,b,c)),k,1,n)+2/(c-a)*Σ(sin(2*k*π*x/(c-a))*
(∫(sin(2*k*π*x/(c-a))*y1_,x,a,b)+∫(sin(2*k*π*x/(c-a))*
y2_,x,b,c)),k,1,n)
EndFunc
```

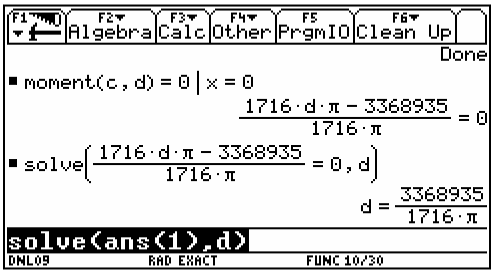


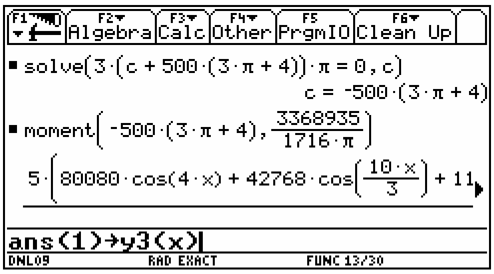


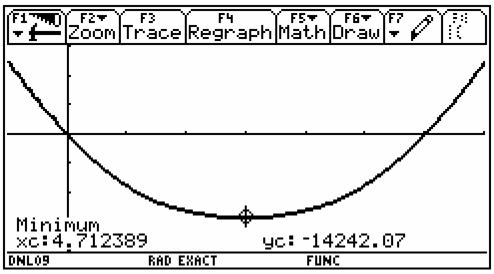












Entdeckungen am Pascal´schen Dreieck mit DERIVE

Günter Scheu, Pforzheim, Germany

DERIVE is an outstanding tool to represent data for a conjecture quickly and systematically in a table or by a graphic. I want to show this on the Triangle of Pascal how to use DERIVE in this sense.

Problems: Try to develop the term for one line in the T.o.P.
Calculate the sum of the coefficients!
What do you notice?
What could be a rule for the sum?
Proof your conjecture for the first 21 rows!
And now for row 100!
Produce a T.o.P with 11 rows! (or more!)
Produce a T.o.P with rows' numbers and the rows's sums!
Produce a T.o.P with a heading!
Give the coefficients modulo 2, modulo 3, modulo k.....
Consider the patterns! Try different values for k!

```
#1:   z(n) ≔ VECTOR(COMB(n, k), k, 0, n)

#2:   z(3) = [1, 3, 3, 1]

#3:   s(n) ≔ ∑(z(n))

#4:   s(3) = 8

#5:   VECTOR(s(n), n, 0, 20)

#6:   [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, 32768,
       65536, 131072, 262144, 524288, 1048576]
```

#7: $[1, 2, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}, 2^{14}, 2^{15}, 2^{16}, 2^{17}, 2^{18}, 2^{19}, 2^{20}]$

#8: $[s(100), FACTOR(s(100))] = [1267650600228229401496703205376, 2^{100}]$

```
#9:   DisplayFormat≔Compressed

#10:  TABLE([z(n),s(n)],n,0,11)
```

#11:

0	[1]	1
1	[1,1]	2
2	[1,2,1]	4
3	[1,3,3,1]	8
4	[1,4,6,4,1]	16
5	[1,5,10,10,5,1]	32
6	[1,6,15,20,15,6,1]	64
7	[1,7,21,35,35,21,7,1]	128
8	[1,8,28,56,70,56,28,8,1]	256
9	[1,9,36,84,126,126,84,36,9,1]	512
10	[1,10,45,120,210,252,210,120,45,10,1]	1024
11	[1,11,55,165,330,462,462,330,165,55,11,1]	2048

```
#12:  Z2b(n):=VECTOR(IF(MOD(COMB(n,k),2)=0,“ 0 “,“   “),k,0,n)

#13:  TABLE(Z2b(n), n, 0, 25)
```

```
#14:
 0  [ ]
 1  [ , ]
 2  [ , 0 , ]
 3  [ , , , ]
 4  [ , 0 , 0 , 0 , ]
 5  [ , , 0 , 0 , , ]
 6  [ , 0 , , 0 , , 0 , ]
 7  [ , , , , , , , ]
 8  [ , 0 , 0 , 0 , 0 , 0 , 0 , 0 , ]
 9  [ , , 0 , 0 , 0 , 0 , 0 , 0 , , ]
10  [ , 0 , , 0 , 0 , 0 , 0 , 0 , , 0 , ]
11  [ , , , , 0 , 0 , 0 , 0 , , , , ]
12  [ , 0 , 0 , 0 , , 0 , 0 , 0 , , 0 , 0 , 0 , ]
13  [ , , 0 , 0 , , , 0 , 0 , , , 0 , 0 , , ]
14  [ , 0 , , 0 , , 0 , , 0 , , 0 , , 0 , , 0 , ]
15  [ , , , , , , , , , , , , , , , ]
16  [ , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , ]
17  [ , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , ]
18  [ , 0 , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , 0 , ]
19  [ , , , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , , , ]
20  [ , 0 , 0 , 0 , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , 0 , 0 , 0 , ]
21  [ , , 0 , 0 , , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , , 0 , 0 , , ]
22  [ , 0 , , 0 , , 0 , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , 0 , , 0 , , 0 , ]
23  [ , , , , , , , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , , , , , , , ]
24  [ , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , , 0 , 0 , 0 , 0 , 0 , 0 , 0 , ]
25  [ , , 0 , 0 , 0 , 0 , 0 , 0 , , , 0 , 0 , 0 , 0 , 0 , 0 , , , 0 , 0 , 0 , 0 , 0 , 0 , , ]
```

```
#16:  Z2_b(n):=VECTOR(IF(MOD(COMB(n,k),2)=1, “1 “, “   “),k,0,n)

#17:  TABLE(Z2_b(n),n,0,10)
```

```
#18:
 0  [ 1 ]
 1  [ 1 , 1 ]
 2  [ 1 , , 1 ]
 3  [ 1 , 1 , 1 , 1 ]
 4  [ 1 , , , , 1 ]
 5  [ 1 , 1 , , , 1 , 1 ]
 6  [ 1 , , 1 , , 1 , , 1 ]
 7  [ 1 , 1 , 1 , 1 , 1 , 1 , 1 , 1 ]
 8  [ 1 , , , , , , , , 1 ]
 9  [ 1 , 1 , , , , , , , 1 , 1 ]
10  [ 1 , , 1 , , , , , , 1 , , 1 ]
```

```
#22:  z3b(n):=VECTOR(IF(MOD(COMB(n,k),3)≠0, "   ", " % "),k,0,n)
#23:  TABLE(z3b(n),n,0,25)
```

```
#24:
 0  [   ]
 1  [   ,   ]
 2  [   ,   ,   ]
 3  [   , % , % ,   ]
 4  [   ,   , % ,   ,   ]
 5  [   ,   ,   ,   ,   ,   ]
 6  [   , % , % ,   , % , % ,   ]
 7  [   ,   , % ,   ,   , % ,   ,   ]
 8  [   ,   ,   ,   ,   ,   ,   ,   ,   ]
 9  [   , % , % , % , % , % , % , % , % ,   ]
10  [   ,   , % , % , % , % , % , % , % ,   ,   ]
11  [   ,   ,   , % , % , % , % , % , % ,   ,   ,   ]
12  [   , % , % ,   , % , % , % , % , % ,   , % , % ,   ]
13  [   ,   , % ,   ,   , % , % , % , % ,   ,   , % ,   ,   ]
14  [   ,   ,   ,   ,   ,   , % , % , % ,   ,   ,   ,   ,   ,   ]
15  [   , % , % ,   , % , % ,   , % , % ,   , % , % ,   , % , % ,   ]
16  [   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   ]
17  [   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   ]
18  [   , % , % , % , % , % , % , % , % ,   , % , % , % , % , % , % , % , % ,   ]
19  [   ,   , % , % , % , % , % , % , % ,   ,   , % , % , % , % , % , % , % ,   ,   ]
20  [   ,   ,   , % , % , % , % , % , % ,   ,   ,   , % , % , % , % , % , % ,   ,   ,   ]
21  [   , % , % ,   , % , % , % , % , % ,   , % , % ,   , % , % , % , % , % ,   , % , % ,   ]
22  [   ,   , % ,   ,   , % , % , % , % ,   ,   , % ,   ,   , % , % , % , % ,   ,   , % ,   ,   ]
23  [   ,   ,   ,   ,   ,   , % , % , % ,   ,   ,   ,   ,   ,   , % , % , % ,   ,   ,   ,   ,   ,   ]
24  [   , % , % ,   , % , % ,   , % , % ,   , % , % ,   , % , % ,   , % , % ,   , % , % ,   , % , % ,   ]
25  [   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   , % ,   ,   ]
```

```
#25:  pa(n,m):=VECTOR(IF(MOD(COMB(n,k),m)≠0,"   "," & "),k,0,n)
#26:  pa_(n,m):=VECTOR(IF(MOD(COMB(n,k),m)=0,"   "," & "),k,0,n)
#27:  bild(m,r):=TABLE(pa(l,m),l,0,r)
#28:  bildNEG(m,r):=TABLE(pa_(l,m),l,0,r)
#29:  [bild(3,10),TABLE(z(n),n,0,10)]
```

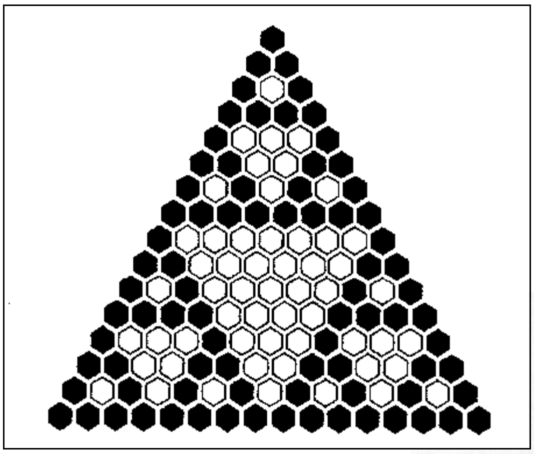
```
#30:
 0  [   ]                                                 ,   0  [1]
 1  [   ,   ]                                                 1  [1,1]
 2  [   ,   ,   ]                                             2  [1,2,1]
 3  [   , & , & ,   ]                                         3  [1,3,3,1]
 4  [   ,   , & ,   ,   ]                                     4  [1,4,6,4,1]
 5  [   ,   ,   ,   ,   ,   ]                                 5  [1,5,10,10,5,1]
 6  [   , & , & ,   , & , & ,   ]                             6  [1,6,15,20,15,6,1]
 7  [   ,   , & ,   ,   , & ,   ,   ]                         7  [1,7,21,35,35,21,7,1]
 8  [   ,   ,   ,   ,   ,   ,   ,   ,   ]                     8  [1,8,28,56,70,56,28,8,1]
 9  [   , & , & , & , & , & , & , & , & ,   ]                 9  [1,9,36,84,126,126,84,36,9,1]
10  [   ,   , & , & , & , & , & , & , & ,   ,   ]            10  [1,10,45,120,210,252,210,120,45,10,1]
```

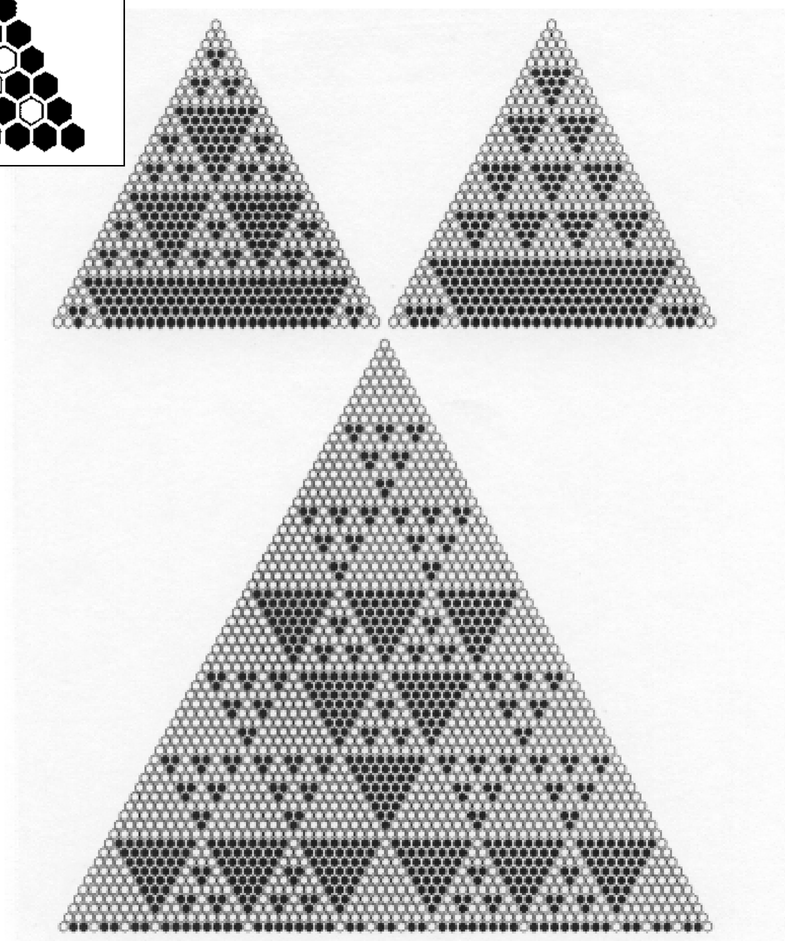
```
#35:  bildNEG(4,27)

  0   [ & ]
  1   [ & , & ]
  2   [ & , & , & ]
  3   [ & , & , & , & ]
  4   [ & ,   , & ,   , & ]
  5   [ & , & , & , & , & , & ]
  6   [ & , & , & ,   , & , & , & ]
  7   [ & , & , & , & , & , & , & , & ]
  8   [ & ,   ,   ,   , & ,   ,   ,   , & ]
  9   [ & , & ,   ,   , & , & ,   ,   , & , & ]
  10  [ & , & , & ,   , & ,   , & ,   , & , & , & ]
  11  [ & , & , & , & , & , & , & , & , & , & , & , & ]
  12  [ & ,   , & ,   , & ,   ,   ,   , & ,   , & ,   , & ]
#36:
  13  [ & , & , & , & , & , & ,   ,   , & , & , & , & , & , & ]
  14  [ & , & , & ,   , & , & , & ,   , & , & , & ,   , & , & , & ]
  15  [ & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & ]
  16  [ & ,   ,   ,   ,   ,   ,   ,   , & ,   ,   ,   ,   ,   ,   ,   , & ]
  17  [ & , & ,   ,   ,   ,   ,   ,   , & , & ,   ,   ,   ,   ,   ,   , & , & ]
  18  [ & , & , & ,   ,   ,   ,   ,   , & ,   , & ,   ,   ,   ,   ,   , & , & , & ]
  19  [ & , & , & , & ,   ,   ,   ,   , & , & , & , & ,   ,   ,   ,   , & , & , & , & ]
  20  [ & ,   , & ,   , & ,   ,   ,   , & ,   ,   ,   , & ,   ,   ,   , & ,   , & ,   , & ]
  21  [ & , & , & , & , & , & ,   ,   , & , & ,   ,   , & , & ,   ,   , & , & , & , & , & , & ]
  22  [ & , & , & ,   , & , & , & ,   , & ,   , & ,   , & ,   , & ,   , & , & , & ,   , & , & , & ]
  23  [ & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & ]
  24  [ & ,   ,   ,   , & ,   ,   ,   , & ,   ,   ,   ,   ,   ,   ,   , & ,   ,   ,   , & ,   ,   ,   , & ]
  25  [ & , & ,   ,   , & , & ,   ,   , & , & ,   ,   ,   ,   ,   ,   , & , & ,   ,   , & , & ,   ,   , & , & ]
  26  [ & , & , & ,   , & ,   , & ,   , & , & , & ,   ,   ,   ,   ,   , & , & , & ,   , & ,   , & ,   , & , & , & ]
  27  [ & , & , & , & , & , & , & , & , & , & , & , & ,   ,   ,   ,   , & , & , & , & , & , & , & , & , & , & , & , & ]

#41:  bildNEG(11,27)

  0   [ & ]
  1   [ & , & ]
  2   [ & , & , & ]
  3   [ & , & , & , & ]
  4   [ & , & , & , & , & ]
  5   [ & , & , & , & , & , & ]
  6   [ & , & , & , & , & , & , & ]
  7   [ & , & , & , & , & , & , & , & ]
  8   [ & , & , & , & , & , & , & , & , & ]
  9   [ & , & , & , & , & , & , & , & , & , & ]
  10  [ & , & , & , & , & , & , & , & , & , & , & ]
  11  [ & ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   , & ]
  12  [ & , & ,   ,   ,   ,   ,   ,   ,   ,   ,   , & , & ]
#42:
  13  [ & , & , & ,   ,   ,   ,   ,   ,   ,   ,   , & , & , & ]
  14  [ & , & , & , & ,   ,   ,   ,   ,   ,   ,   , & , & , & , & ]
  15  [ & , & , & , & , & ,   ,   ,   ,   ,   ,   , & , & , & , & , & ]
  16  [ & , & , & , & , & , & ,   ,   ,   ,   ,   , & , & , & , & , & , & ]
  17  [ & , & , & , & , & , & , & ,   ,   ,   ,   , & , & , & , & , & , & , & ]
  18  [ & , & , & , & , & , & , & , & ,   ,   ,   , & , & , & , & , & , & , & , & ]
  19  [ & , & , & , & , & , & , & , & , & ,   ,   , & , & , & , & , & , & , & , & , & ]
  20  [ & , & , & , & , & , & , & , & , & , & ,   , & , & , & , & , & , & , & , & , & , & ]
  21  [ & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & , & ]
  22  [ & ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   , & ,   ,   ,   ,   ,   ,   ,   ,   ,   ,   , & ]
  23  [ & , & ,   ,   ,   ,   ,   ,   ,   ,   ,   , & , & ,   ,   ,   ,   ,   ,   ,   ,   ,   , & , & ]
  24  [ & , & , & ,   ,   ,   ,   ,   ,   ,   ,   , & , & , & ,   ,   ,   ,   ,   ,   ,   ,   , & , & , & ]
  25  [ & , & , & , & ,   ,   ,   ,   ,   ,   ,   , & , & , & , & ,   ,   ,   ,   ,   ,   ,   , & , & , & , & ]
  26  [ & , & , & , & , & ,   ,   ,   ,   ,   ,   , & , & , & , & , & ,   ,   ,   ,   ,   ,   , & , & , & , & , & ]
  27  [ & , & , & , & , & , & ,   ,   ,   ,   ,   , & , & , & , & , & , & ,   ,   ,   ,   ,   , & , & , & , & , & , & ]
```





[1] G.Scheu, Arbeitsbuch Computer-Algebra mit DERIVE, Dümmler
[2] Peitgen a.o., Fractals for the Classroom, Springer
[3] B.Bondarenko, Generalized Pascal Triangles, Tashkent, Fan, 1990 (in Russian)

DERIVE's NEW FEATURES (last versions)

1. The new function RANDOM implements a pseudo-random number generator. Several functions using the random number generator are defined in the utility file MISC.MTH. (RANDOM_SIGN, RANDOM_POLY, RANDOM_VECTOR, RANDOM_MATRIX).

2. The functions MOD, FLOOR, GCD, and LCM formerly defined in the utility file MISC.MTH are now efficiently defined in DERIVE.EXE. FLOOR is now used to efficiently define the functions CEILING(m,n) (= [m/n]) and ROUND(m/n) in MISC.MTH.

3. Number theory functions for finding the nth Fibonacci, Bernoulli, and Catalan numbers are now defined in MISC.MTH.

4. The function APPEND formerly defined in the utility file VECTOR.MTH is now efficiently defined in DERIVE.EXE.

5. For convenience, the function FIT for finding the best least-squares fit to a parameterized expression now accepts two arguments: a label vector and a data matrix.

6. The new function GOODNESS_OF_FIT defined in MISC.MTH returns the goodness of the fit of the expression returned by the built-in function FIT.

7. The Gamma function can now find a numerical approximation for complex as well as real arguments.

8. The Riemann zeta function formerly defined in the utility file ZETA.MTH is now defined in DERIVE.EXE. In exact mode, ZETA(s) returns an exact closed form value if one is known. In approximate mode, it returns an approximate answer. Note that ZETA(s) is returned as the sum from k=1 to infinity of 1/k^s, if s>1.

9. Functions for approximatig modified Bessel functions of the first and second kind are now defined in the utility file BESSEL.MTH.

10. The utility files ODE1.MTH and ODE2.MTH now include general purpose functions that automatically choose the appropriate method for solving a differential equation. Functions are provided for finding a specific solution when given an initial condition or boundary value, and for finding a general solution in terms of a symbolic constant c.

11. The utility file RECUREQN.MTH now has single-step functions for finding general and boundary value solutions to constant coefficient second-order linear difference equations.

12. The new Transfer Load daTa command loads numeric data files produced by other mathematical software programs.

13. Utility, demonstration, and numeric data files can now be loaded directly from DOS using a command line switch.

14. The new Transfer Save C command saves expressions in a C language file in a form suitable for including in C programs.

15. When a Plot command is issued from an Algebra window, you are now given the opportunity to select the location for the new plot window. The choices are "Beside", "Under", or to "Overlay" the Algebra window.

16. Powers of system or user-defined functions can be entered by putting the power immediately after the function name.

17. The new appendix "Line Editing Commands" in the DERIVE User Manual summarizes the line editing commands available when authoring an expression. Also the appendix "Additional Resources" includes several new books based on DERIVE.

In addition to the new features this version incorporates numerous internal improvements to DERIVE's factorizor, equation solver, and algebraic simplifier.