Introducing the concept of integration with the TI-92

Wolfgang Pröpper
Nürnberg, Germany,

How was INTEG() originated?

At the wonderful Fun-in-Learning-Mathematics conference last summer in Sweden Josef Böhm gave me a program group called "Riemann". He told me it would be of excellent use for introducing the concept of definite integral. He himself got the basic idea from Jose Santonja from Spain, who sent him a contribution to the DERIVE News Letter.

Already on my way home I started practising with that program because I already had used Josef´s and Terence Etchells´ utilities for DERIVE [1] with great success (and after modifications that came nearer to my taste) in my lessons. The hope to be able doing those things with the TI-92 inspired me. However I was disappointed, when I saw, that the program was awkward in the input, included dubious methods and and crashed down again and again. Because of that I firstly put it aside.

Every confirmed DERIVEr knows that in most cases this is not the end. When looking for the reason why the program had such an unfriendly behaviour, I found some programming techniques which let it seem worth while to continue working on that project. So by the way I got an imagination how a program dealing with Tiemann´s concept of integration which was simple on the input side and save when running could look like. October 1997 I sent Josef a first version with the name INTEG(). A busy exchange of thoughts started which on the one hand led to the conclusion to found an Austrian-German joint venture and on the other hand resulted in a steady growth, so that INTEG() grew thick and thicker. At this time it includes 23 modules whose code occupies about 27.000 bytes of the TI-92.
In this talk I will show you a general view on the most important functions of INTEG(). Questions on how to use it in a classroom and further informations will be discussed in a workshop which Josef Böhm and I are presenting.

How is INTEG() working?

After starting the program an an introductory screen opens (Fig. 1), which shows the user how he can possibly get help for operating the program, where redy to go examples are found and how the parameters the program needs are being entered. We want to talk through a realistic example and open the Parameter Menu (Fig. 2) by pressing (F2). The first time one would enter all parameters, namely the function´s term, the plotting range, the range of integration and the number of strips. Later on each parameter can be changed without having to run through the whole input procedure again. With item (7) one can switch between the "graphical" mode (which is a default) and the "numerical" mode (which comes automatically if the range of integration or the number of strips contaim variables).

We first look at the function $f(x) = x^3 - 4x^2 + 4x + 1$. It should be plotted in the range $-0,5 \le x \le 4,5$ and $-0,5 \le y \le 4,5$. The range of integration should be [0 .. 3] and it should be separated by 10 strips. When entering the data some checks are made. For instance improper plotting ranges or foolish inputs on the number of strips will be rejected. After a correct input the screen shows all date entered (Fig. 3).

After opening menu ✍(F3) one can select from a great variety of methods (Fig. 4). The first 6 methods are based on aequidistant strips. Simpson´s Rule and the Pulcherima method (it is a method which was discovered by C.F.Gauss and got this most beautiful name in Austria) are in this case interpreted with strips of different width. At method 9 the width of the strips is a geometric sequence while in method 11

the strips are generated by random. Method 10 finally is the well known random rain where the drops falling within the interesting area are compared with the total number of raindrops.
When strting one of those methods in the graphic mode the screen is firstly split vertically in a ratio 1:2. In the right screen a picture consisting of the graph of the function and the chosen pattern of strips (or the random dots) is generated. After that the calculated sum appears in the left screen in exact mode and as decimal approximation (with 3 significant digits). Figs. 5 through 8 show some typical cases.

In certain cases the exact value of the sum can only be shown partly in the small output window when working in graphic mode. In this case one can switch by (F2) (7)✍ to the numerical mode. Now the whole screen is available for the calculated approximation (Fig. 9). Pressing the (P)✍ key (for ✍(P)rocessing) opens a popup menu with the most important processing functions of the TI-92 (Fig. 10). Now the resulting terms can be manipulated in various ways.
However much more interesting is the numerical mode when using variables for the number of strips and/or the integration bounds instead of numbers. (The input check only allows "n" for the number of strips and "a" or "b" for the bounds.) Now the expression of the sum can be processed with the given methods. In many cases one can reach simplification up to the definite integral (Fig. 11 through 14).

The other two menus work independently from the setting "graphical" or "numerical".

In menu ✍(F4) two comparing procedures for the different methods, in one case with a graphical completion are offered (Fig. 15). These comparisons only make sense when integration bounds and the number of strips are numerical.
When comparing "all methods" ((F4) (1)) the calculated approximations have an output with up to 6 significant digits on the screen (Fig. 16). Among the methods compared the Monte Carlo method is missing because the number of raindrops, which is internally managed like the number of strips, will normally be too little. For the "geometric sequence" method the sum will be calculated only if the integration range is totally positive.
When comparing "selected methods" ((F4) (2)) the user can choose several methods an can afterwards enter a range for the number of strips for which those methods are executed. First a popup menu (Fig. 17) opens, in which the desired methods get a click when choosing them. With the command (A)✍`:Calculate` the menu is closed. The following dialog (Fig. 18) asks for the starting and final value of the number of strips and for the steps. The output comes in approx mode with 3 significant digits in a matrix (Fig. 19). Because of the limited size of the TI-92´s screen there is not enough space for the matrix when choosing more than 4 methods and/or more than 6 different numbers of strips. When exceeding those limits the matrix can anyhow be inspected after closing the program in the Home Screen or in the Data/Matrix Editor. (Understandably the computation time is growing rapidly, so one should think about it bevor exceeding the limits mentioned above.)
However when the matrix is calculated one can get a graphical representation of the numerical values straight afterwards by pressing (F4) (3). The x-axis is the number of strips. The methods are represented by different characters (Fig. 20). In case the matrix has not benn calculated immediately bevore calling this item the program branches to the routine (F4) (2) independently.

Finally menu (F5) offers integralfunctions, that means, functions of the form $F_u(x) = \int_u^x f(t)dt$ with constant lower bound u. These functions can, with different values for u be outputted analytically, which means as an algebraic term or grahpically. In both cases the program first asks for the lower bound (only numerically). The choice of (F5) (2) separates the screen, like at the methods in the graphic mode, vertically in ratio 1:2. On the right side the graph of the aktual function is plotted in dots and

afterwards the integralfunction $F_u(x)$ is plotted. This plotting of integralfunctions can be reapeted by pressing the (N) key and entering an new lower bound. By this one gets a family of curves whose members only differ by an additive constant (Fig. 21). However, pressing the (A) key the algebraic term of the integralfunction which was plotted last will be shown (Fig. 22).
In case one has made a decission for the analytical form first by pressing the (1) key after the (F5) key one can at any time switch to the graphical display by pressing (G) or select an new integralfunction by pressing (N). In both modes (E) closes this procedure and the user returns to the main screen of `INTEG()`.

The function key (F6) shows a set of examples. It saves the user from entering parameters in menu (F2). At the first call a little information text is shown. The displayed example can be accepted with (ENTER). By pressing (ESC) the procedure is passed without any change an any other key leads to the next example.

`INTEG()` should always be closed by (F1) (2). In case the (ENTER) key is pressed, `INTEG()` branches into a finishing routine that deletes all variables generated during running the program and tells the TI-92 the settings from before the program was started. However leaving `INTEG()` with an (ESC), all variables and setting are being kept. Especially the variable `erg` contains the last result that was calculated in menues (F3) and (F4). The variable `erl` contains the last shown result from (F3). With these variables one can continue working in the Home Screen (or, if `erg` is the matrix from (F4) in the Data/Matrix Editor). The final housekeeping is done by calling `INTEG()` once more and immediately leaving it by pressing (F1) (2) (ENTER).

Conclusion

As mentioned in the beginning I wanted to sketch the performance of `INTEG()`. But who wants to work with this program will first have to train on it to get to know all methods in their different views. Then however it can well be used in lessons as a medium. If a whole class is equipped with TI-92s one can also give tasks to the pupils. With appropriate instructions they can with this program be lead to the concept of integration.

For interested people: `INTEG()` can be downloaded from the TI-92 page of my school. The URL is http://www.kolleg.nuernberg.de/ti92.htm. It comes as a selfextracting archive called integ.exe which contains a group file integ.92g and a WinWord 6.0 documentation called integ.doc.

For inquiries, stimulations or vituperations Josef Böhm and I can be bombed with a lot of emails:
(Josef.Boehm@bboard.blackbox.or.at bzw. w.proepper@wpro.franken.de)

I want to thank Josef Böhm for his ideas and his critical companionship during the programming and I also want to thank you for your intensive listening and your patience.

Literature:

[1] Josef Böhm, Terence Etchells: various contributions in DNL #7 and #8, Würmla 1992

Illustrations to the text

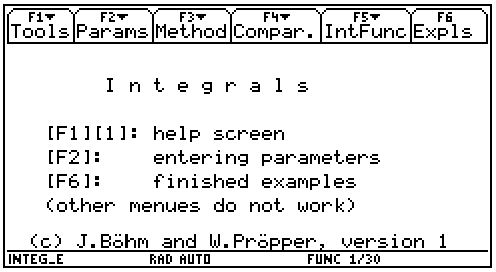

Fig. 1 The introductory screen

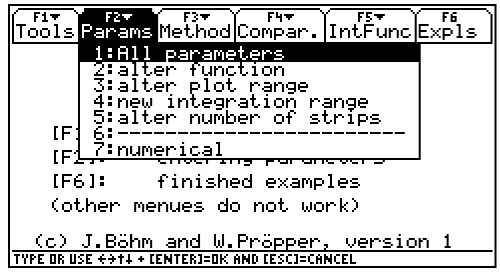

Fig. 2 The parameter menu

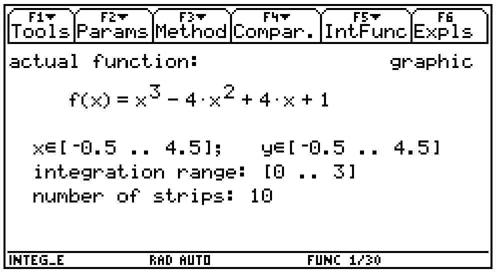

Fig. 3 Mainscreen with parameters

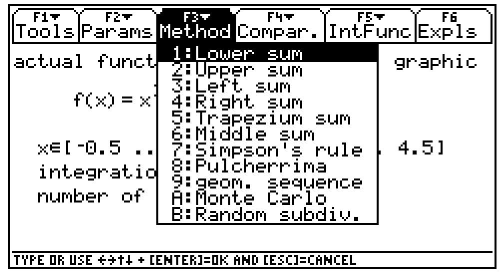

Fig. 4 The methods menu

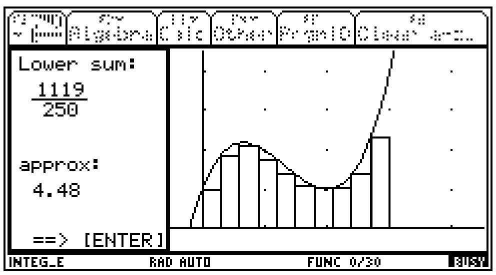

Fig. 5 Lower sum to Fig. 3

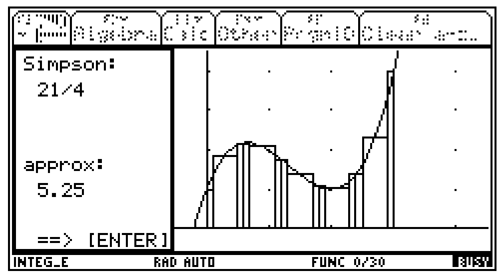

Fig. 6 Simpson´s method with n=5

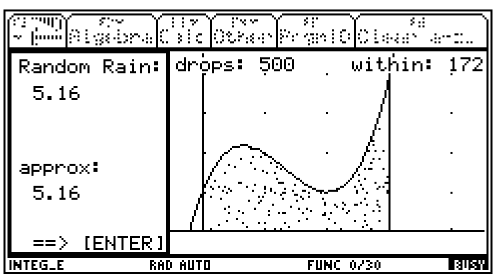

Fig. 7 500 raindrops

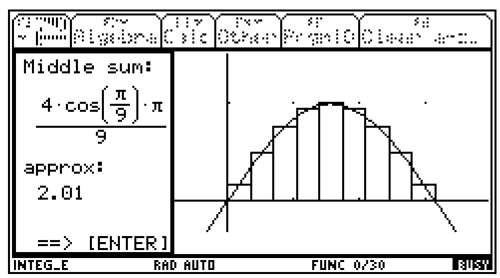

Fig. 8 Midsum to f(x)=sin(x)

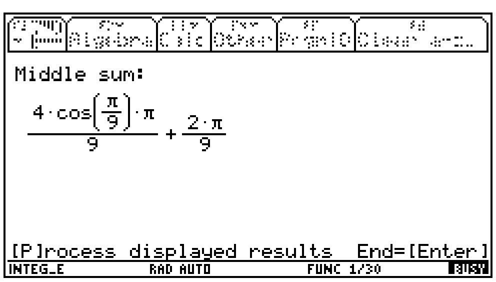

Fig. 9 like Fig. 8, in num. mode

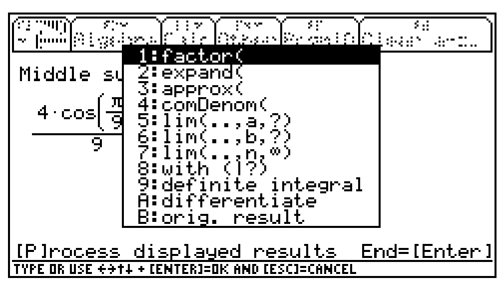

Fig. 10 (P)rocessing menu

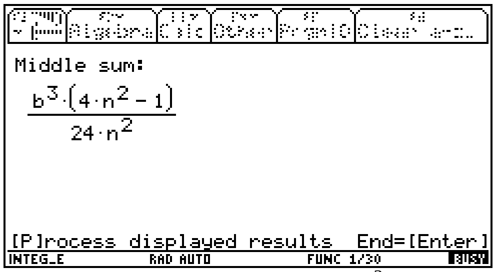

Fig. 11 MSum to $f(x)=½x^2$ in [0..b]

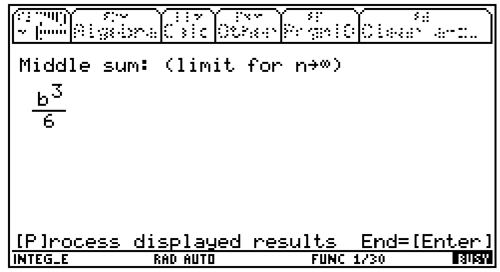

Fig. 12 Limit to Fig. 11 for n→∞

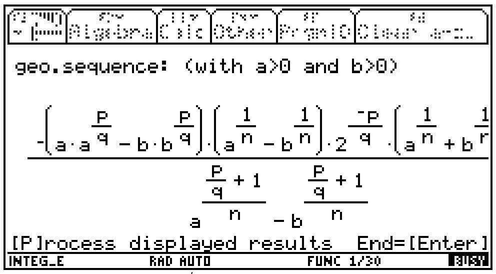

Fig. 13 $f(x)=x^{p/q}$ in [a .. b] with geo. seq.

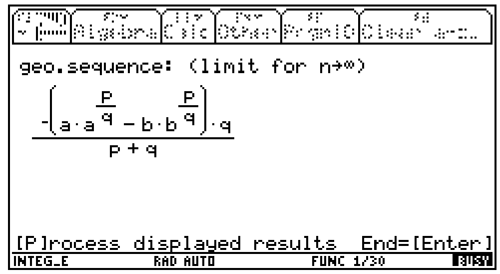

Fig. 14 Limit only with geo. sequence

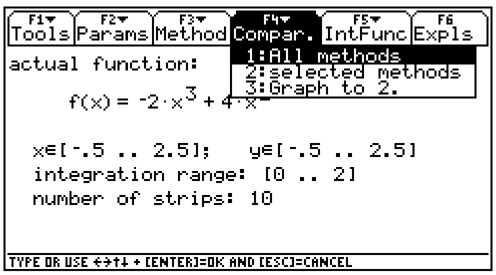

Fig. 15 Comparison menu

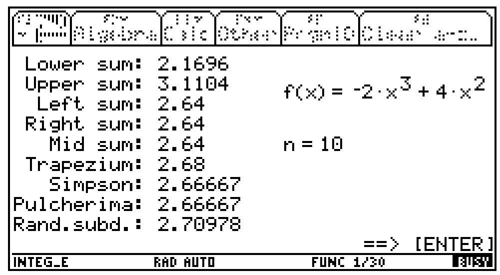

Fig. 16 All methods (s. Fig. 15)

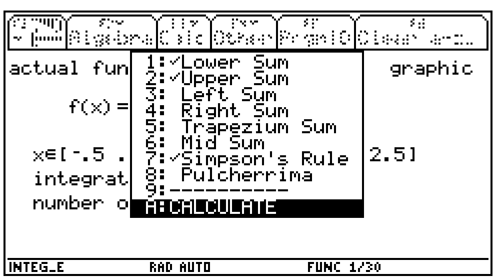

Fig. 17 Lower, Upper Sum, Simpson

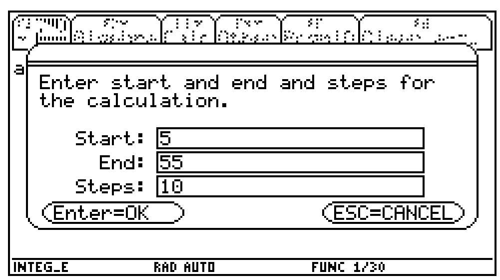

Fig. 18 for n = 5, 15, .., 55

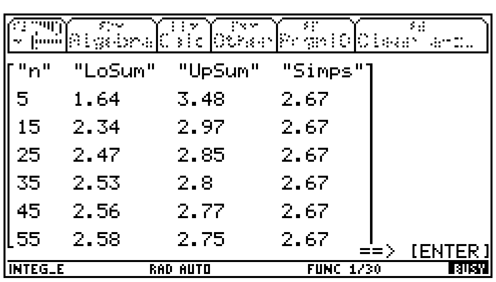

Fig. 19 Does Simpson yield exact value?

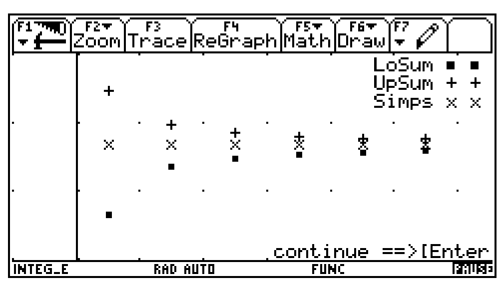

Fig. 20 Graphical interpretation

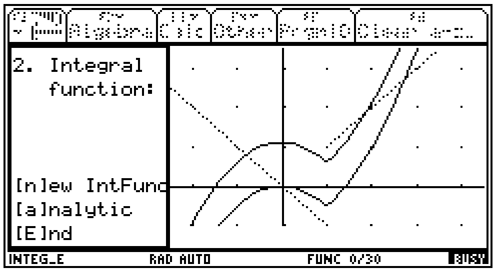

Fig. 21 Two integralfunctions to the unsteady func. f(x) = |x–1| + sign(x–1)

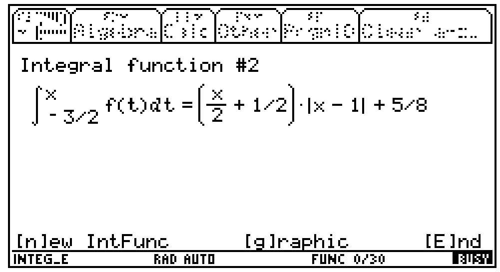

Fig. 22 Analytical representation of $\int_{-3/2}^{x} \left[|t-1| + \text{sign}(t-1)\right] dt$