

A worksheet of the engineering calculator as a structured flowchart

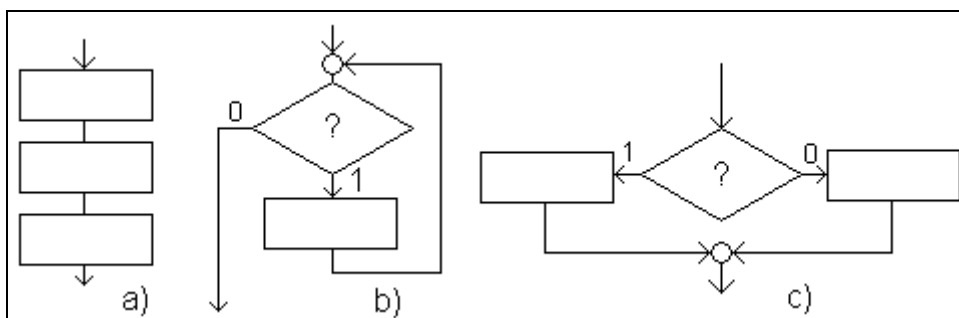
Valery Ochkov (http://twt.mpei.ac.ru/ochkov/v_ochkov.htm)

The author considers in the article problems of input elements of programming directly into documents created in so-called engineering calculators, the most famous representative of which is the program Mathcad.

Calculation in Mathcad worksheet [1] is «naturally ordered»: from left to right and from top downward. But solutions of some problems require changes of this «linear» order. For example, the method of successive approximations often used in engineering and technical calculations, requires that a group of operators must be calculated several times unless a certain condition is executed. In this case some Mathcad users duplicate the group of operators several times in the worksheet. The other (more advanced) users resort to programming tool, i.e. return to that formerly left. After all, Mathcad package was developed for those who didn't want or couldn't use programming languages to solve their own problems.

The technique proposed below enables users to solve this contradiction and enhance capability and flexibility of solving tools in Mathcad environment retaining simplicity and clearness of the worksheets themselves.

It is known that an algorithm of any complexity can be implemented using structure controls of sequence of operations (process), iteration, and decision¹. Fig. 1 shows these structure controls as the flowcharts.



¹ This follows from the main structure theorem attributed to E.Dijkstra (see http://en.wikipedia.org/wiki/Edsger_W._Dijkstra). Yet, the third control, decision, of this trinity (process, iteration, and decision) is superfluous as it can be changed for two reiterations, two cycles executed once or never.

Figure 1. Flowcharts of the main structure controls a) process, b) iteration (pre-test loop), c) decision

Algorithm constructions process, iteration, and decision can be represented as flowcharts and as so called structured flowcharts² [2], see Fig. 2.

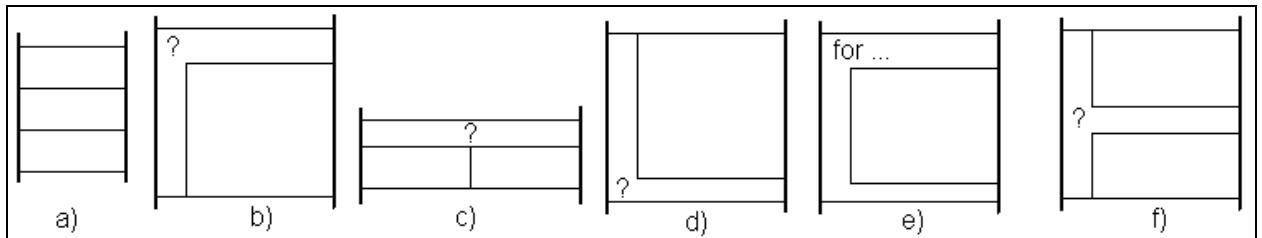


Figure 2. Structure symbols: a) process, b) pre-test loop c) decision, d) post-test loop e) FOR loop f) N-plus-a-half loop

For programming convenience additional structure controls are introduced into real languages: post-test loop³ (p. d on Fig.2), for loop (p. e on Fig.2), N-plus-a-half loop (p. f on Fig.2), etc.

Thus, in addition to pre-test loop there is a FOR loop in Mathcad. Besides, Mathcad allows users to prematurely exit WHILE and FOR loops with operators `break` (returns control to the first statement after the loop) and `continue` (returns to the first loop statement). There were rejected operators of conditional GOTO statement (`if... goto...`) and unconditional one (`goto label`) in Mathcad that coincides the idea of structure programming but creates a certain inconvenience in a practical use⁴. Operators `break` and `continue` (and operator `return` halting execution of a program) play role of certain compromise here. Programming with use of labels (a circle having two inward arrows and one outward, see Fig. 1) and operators of conditional statement (a rhomb having question-mark inside, see Fig. 1) is a nature of flow

² In [4] all the programs-graphics were framed with structured flowcharts. The history of structured flowcharts (Nassi-Shneiderman flowcharts, Structograms) is described here: <http://www.cs.umd.edu/hcil/members/bshneiderman/nsd>.

³ Repeat construction is a pre-test loop (see Fig. 1 and 2). A pre-test loop is more «major» than a post-test loop because the body of the pre-test loop may run not even once, but the body of the post-test loop should be iterated once at least [2].

⁴ This inconvenience generally reveals if it is required to transform programs written in languages in which labels are basic tool for algorithm branching into the Mathcad program.

charts. But structure programming (in other words, programming without GOTO) is better to represent by structured flowcharts shown on Fig. 2.

In a Mathcad worksheet it is possible to create only *process* structure control: as was noted above, calculating operators are executed sequentially from left to right, downward. To change this order it is required to introduce a «foreign» element, programming, into the worksheet. Although, we can accept that other structure controls⁵ will be developed in the worksheet. Imagine, if one clicks the buttons `while` or `until` in the new Flowchart tool box (see Fig. 3) the Mathcad worksheet will become lined horizontally and vertically representing structure of the pre-test loop shown in point b of Fig. 2. The left and the right lines will bound the Mathcad worksheet⁶ itself and other horizontal and vertical lines (a user can drag them from place to place) will fix a header and a body of the pre-test loop; the header of the loop will contain the keyword `while` or `until`⁷.

⁵ This has been already introduced in some way: it is possible to insert function *if* into Mathcad worksheet executing decision construction. Although, a significant limitation gets in the way – the arguments of the function *if* should be only a pair of operators forming two branches of the decision construction.

⁶ These lines present in any Mathcad worksheet. The left vertical line is the left border of the screen. The right vertical line really crosses downward an operational field of a Mathcad worksheet noting the right margin of print area.

⁷ The keyword `until` is usually used for the post-test loop. But we accept that the keyword *until* can be used in the pre-test loop (as it is executed in BASIC) assuming that `until x` is equivalent to construction `while not(x)`.

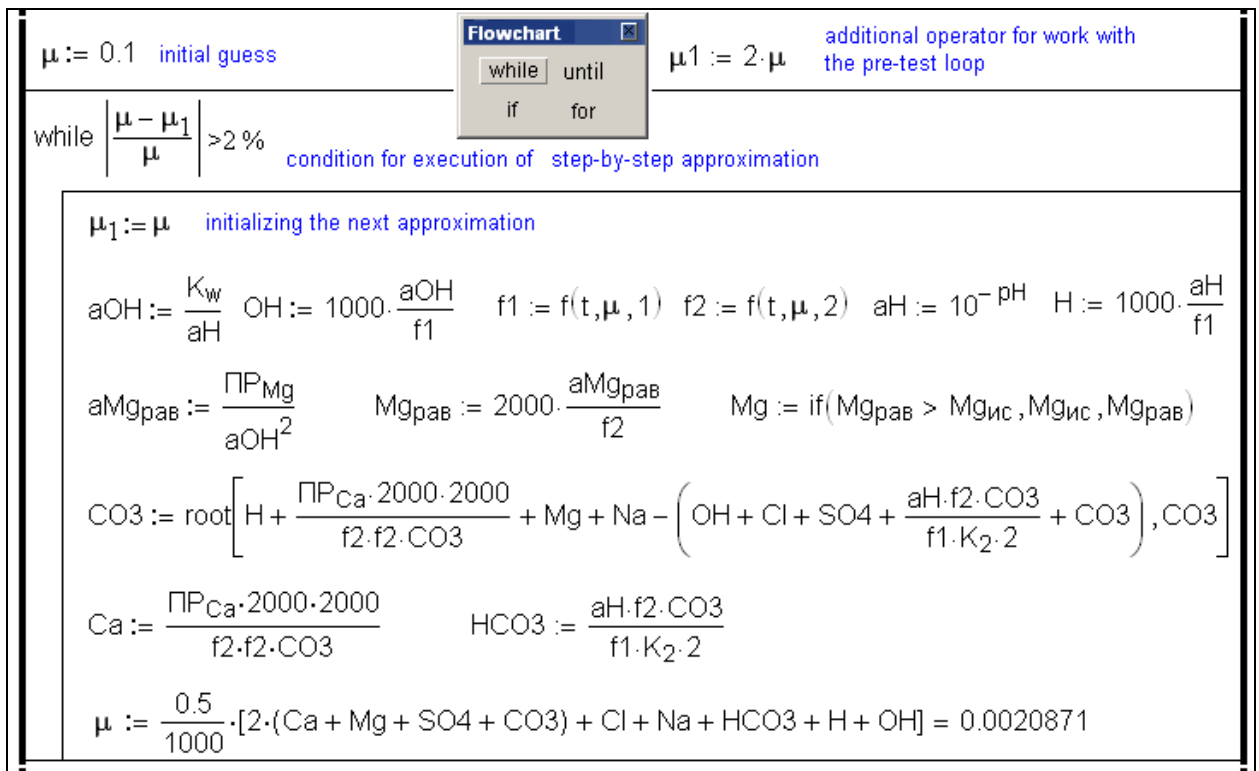


Figure 3. While loop in the modified Mathcad environment

The areas below, delineated in that way, may include: first, Boolean expression controlling the loop (it is written into the header of the loop, into the Γ -type area), and second, usual Mathcad operators forming the body of the loop (inner rectangular area). Thus, Fig. 3 shows method of successive approximations noted above created in such modified Mathcad: the initial guess is given and it is verified in the loop until two latest approximations differ moderately from each other. The header of such loop can additionally contain only comment (as shown on Fig. 3) but the body can comprise any Mathcad operator including evaluation operator (=), see the last operator in the loop body, shown on Fig. 3. As the need arose, this operator helps to debug a program by viewing the intermediate results obtained during run of the loop⁸. For this it is necessary to expand the list of commands applied to separate operators with the command *Pause* which interrupts calculation in this operator and it could be continued by pressing button F9.

⁸ Now to do it in Mathcad a user should insert functions *trace* or *pause* returning values into the special Trace Window that is inconvenient.

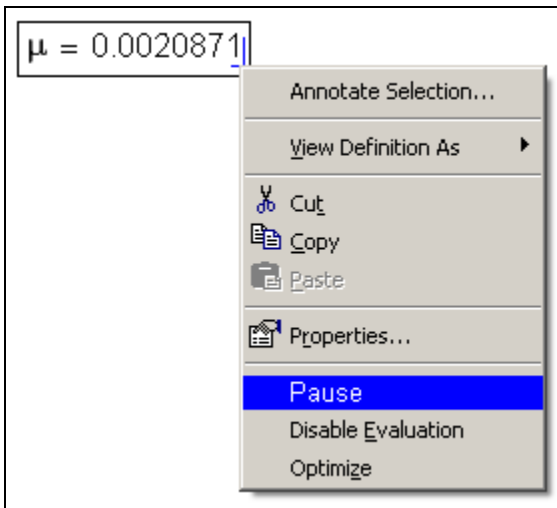


Figure 4. Command (switcher) *Pause* in the list of commands applied to the separate operator.

It will be also possible to place plots into the body of such loop, therefore to create animation and see changing various lines on the plot during run of the loop.

The problem shown on Fig. 3 (method of successive approximations) is better to solve not by WHILE loop but using more generalized construction, a N-plus-a-half loop⁹, see Fig. 5.

$\mu := 0.1$ <i>initial guess</i>	
$f1 := f(t, \mu, 1) \quad f2 := f(t, \mu, 2) \quad aH := 10^{-pH} \quad H := 1000 \cdot \frac{aH}{f1}$ $aOH := \frac{K_w}{aH} \quad OH := 1000 \cdot \frac{aOH}{f1} \quad aMg_{рав} := \frac{PP_{Mg}}{aOH^2} \quad Mg_{рав} := 2000 \cdot \frac{aMg_{рав}}{f2}$	
<i>validity checking</i> if $Mg_{рав} > Mg_{ис}$	
<i>yes</i> $Mg := Mg_{ис}$	<i>no</i> $Mg := Mg_{рав}$
$CO3 := \text{root} \left[H + \frac{PP_{Ca} \cdot 2000 \cdot 2000}{f2 \cdot f2 \cdot CO3} + Mg + Na - \left(OH + Cl + SO4 + \frac{aH \cdot f2 \cdot CO3}{f1 \cdot K_2 \cdot 2} + CO3 \right), CO3 \right]$ $Ca := \frac{PP_{Ca} \cdot 2000 \cdot 2000}{f2 \cdot f2 \cdot CO3} \quad HCO3 := \frac{aH \cdot f2 \cdot CO3}{f1 \cdot K_2 \cdot 2}$ $\mu1 := \frac{0.5}{1000} \cdot [2 \cdot (Ca + Mg + SO4 + CO3) + Cl + Na + HCO3 + H + OH]$	
until $\left \frac{\mu - \mu1}{\mu} \right \leq 2\%$ <i>condition for termination of the approximations</i>	
$\mu := \mu1$ <i>initializing the next approximation</i>	
$\mu = 0.0020871$ <i>the obtained value of step-by-step approximations</i>	

Flowchart

while until

if for

⁹ To put it into operation in current Mathcad one inserts an infinite loop into the calculation (while 1...) which contains operator *break if...*, that is again unnatural, so to speak.

Figure 5. A N-plus-a-half loop

To insert any loop into the calculation (except for *for*-loop as it has got its own button) a Mathcad user should click button `while` or `until` on the toolbar *Flowchart* and Mathcad will represent the construction shown in point f on Fig. 2. If a keyword `while/until` is dragged upward it will be transformed into a pre-test loop (point b on Fig. 2), to do it downward will return the construction into post-test loop¹⁰ (point d on Fig. 2).

Using the same technique makes it possible to introduce decision construction, see Fig. 6 showing calculation of the tax by a progressive scale. The upper part of the Fig. 6 shows the calculation with the help of traditional programming tools (it represents existing Mathcad 14 toolbar); the middle part of the Fig. 6 shows the calculation with the help of programming tools of Mathcad Prime toolbar¹¹; the bottom part represents it using proposed decision structure control. By the way, this construction has been inserted into the loop shown on Fig. 5 and it has been put into operation using built-in Mathcad function `Mg :=if (...)` on Fig. 3.

¹⁰ It is acceptable to insert several operators `while/until` into the loop creating several N-plus-a-half loops. Also, it will be possible to run some operators (process construction) before loop termination. All these methods improve flexibility of programming. But there is an apprehension: a worksheet would become less clear for understanding.

¹¹ An additional vertical line appeared on Mathcad Prime right of the operators of the program or its individual modules. This feature of "closer" of the program Mathcad Prime to structural diagrams.

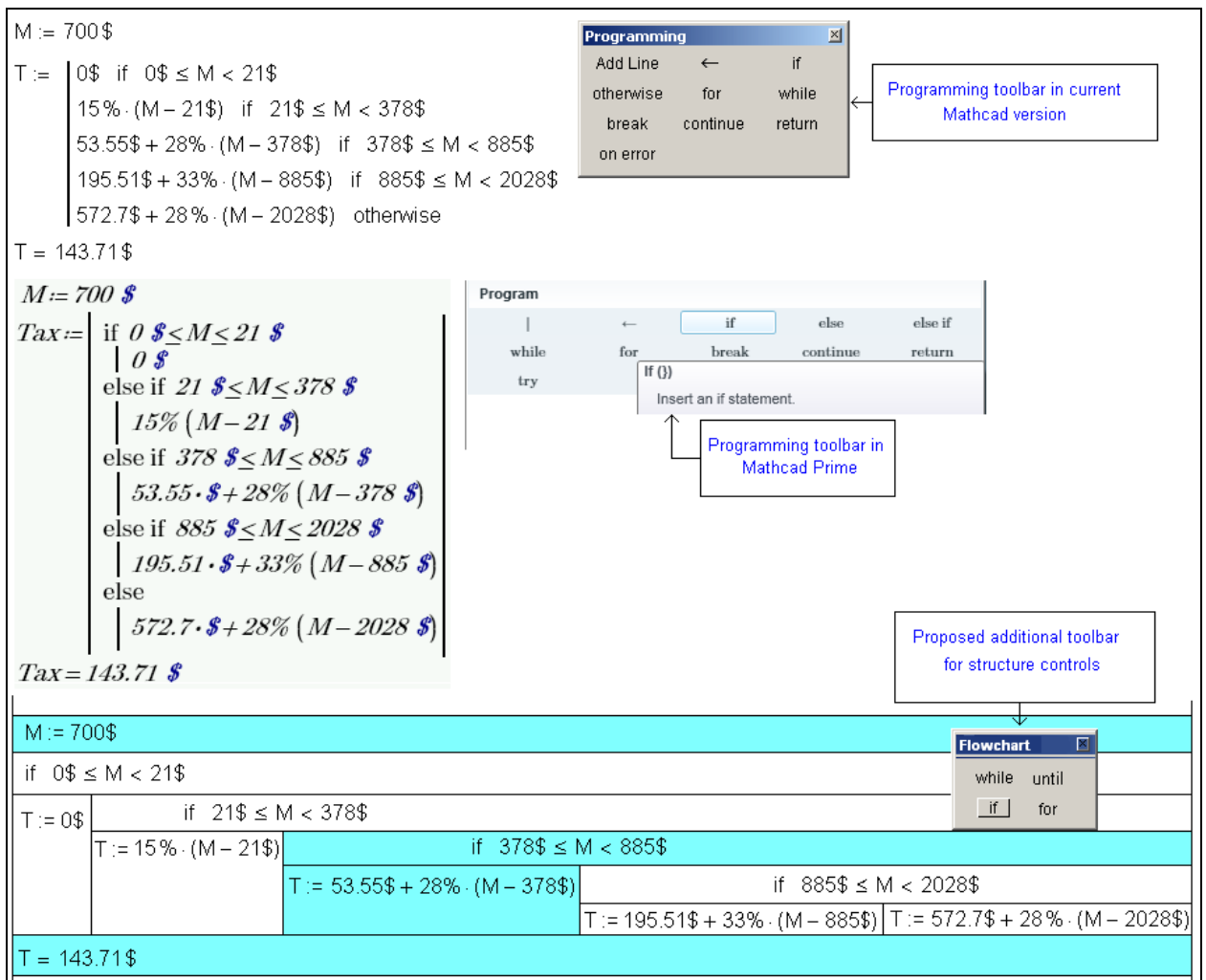


Figure 6. Two variants of the decision construction

Fig. 6 shows that it will be possible to highlight elements chosen for calculation for the given case ($M := 700\$$) developing the decision construction.

The proposed structure controls can be used for creating the user functions in Mathcad. Thus, Fig. 7 shows the function named $zero$ returning null of another function named y by bisection method. The upper part of Fig. 7 represents creating the function by the traditional Mathcad tool, the bottom part shows that by the proposed technique.

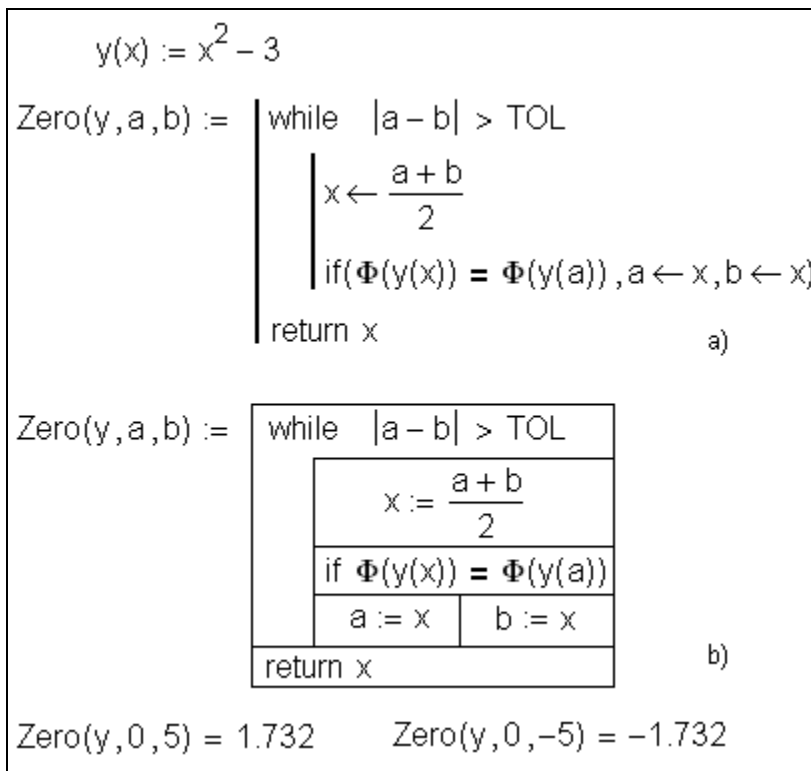


Figure 7. Two methods of creating the user function in Mathcad a) existing method b) proposed method

Today this technique that enables us to increase flexibility and capability of the engineering calculators maintaining its natural notation is being adopted in SMath project (see www.smath.info)

References:

1. Ochkov V.F. Mathcad 14 for students and engineers: Russian version. BHV-Petersburg, 2009 (in Russian)
2. Nassi, I. and Shneiderman, B., Flowchart Techniques for Structured Programming, SIGPLAN Notices 8, 8 (August, 1973 - <http://www.cs.umd.edu/hcil/members/bshneiderman/nsd/1973.pdf>).
3. Ochkov V.F. Odd man out or About basic structure controls // Software products and systems, № 1, 1989 (in Russian)
4. Ochkov V.F., Puhnachev Yu.V. 128 advices for a first-time programmer. Moscow, Energoatomizdat, 1992 (in Russian)