BULETINUL Universităţii Petrol – Gaze din Ploieşti	Vol. LXIII No. 2/2011	89 - 97	Seria Tehnică

Easy-to-use Mathcad Computer Methods for Interconversion of Petroleum Distillation Curves. Part 3. ASTM - TBP (True Boiling Point) Conversion Using Composed Functions

Liviu Filotti

Universitatea Petrol-Gaze din Ploiesti, Bd. Bucuresti 39, Ploiesti,
e-mail : lfilotti@upg-ploiesti.ro; lfilotti@yahoo.com

Abstract

The procedures recommended by API [2] for the interconversion of ASTM D 86 and TBP (True Boiling Point) distillation curves are transposed into Mathcad computation platform. Herein presented Mathcad version for both direct, ASTM to TBP, and backward, TBP to ASTM, curves conversions are more concise, easier to use and of a more general application than versions given in previous two parts [1a, b]. In the Mathcad version given in this third part the relations of the API procedure are written in Mathcad as functions and composed functions, in a sequence which finally allows use of the procedure as a single-variable function by a unique call. The utilization of Mathcad-written procedures is illustrated with examples.

Key words: *ASTM D 86, TBP, distillation, interconversion, Mathcad, petroleum*

Introduction

Main goal of this part of the work is to provide a more compact Mathcad implementation of the API (American Petroleum Institute) procedures for the ASTM D 86 and TBP distillation curves interconversion, Mathcad version meant to be of a more convenient use and without the drawbacks of previous versions [1a, b]. As it will be shown, it is possible to employ the herein presented Mathcad version of the API procedure with a unique call of a function of a single variable which is merely the initial (given) distillation curve to be converted. It will thus be possible to use present Mathcad version for repeated tasks or to include it in (Mathcad) programming modules.

Relations and parameters needed for the interconversion of ASTM (ASTM D 86) and TBP (True Boiling Point) distillation curves according API procedure have been given in [1a], hence they are no more shown explicitly here. API procedure is also called of Daubert [3], since it is essentially based on Daubert work [4], hence it will be sometimes referred to in the present paper as API-Daubert procedure, too.

Relations and values of parameters needed in the interconversion procedure have been adapted from the original API work [2] by the present author, in order to be used with temperatures in degrees centigrade (Celsius, °*C*).

Mathcad Implementation of API Procedures for ASTM – TBP Distillation Curves Interconversion. Using Composition of Functions

Direct (ASTM-to-TBP) conversion

The main part of the worksheet with the current version of the Mathcad implementation of the API method for the estimation of TBP curve from ASTM distillation curve is given in Fig. 1. Like for previous versions, we have sought to give to our Mathcad-written procedure a layout which allows an easy recognition of the original method as published in the API book [2]. Mnemonic notations are employed for various variables and functions names throughout the Mathcad worksheet (legend of notations and acronyms at the end of the paper).

Characters in text zones are in italics (most of text in Arial font, too), in order to clearly differentiate text from calculation areas.

In first part of the worksheet and before the portion shown in Fig. 1, there is the zone with assignments of parameters used in the method, zone which is no more shown since it is same as in [1a]. As established in [1a, b] and in agreement with the API-Daubert procedure, the distillation curve is represented by a 7×2 matrix ("DC" in Fig. 1), that is a matrix with 7 rows and 2 columns, the first column with the % volume of distillate employed by the API method (0, 10, 30, 50, 70, 90 and 100 % vol.) and 2nd column with the corresponding temperatures.

We mention that even if same relations and coefficients are used, in some references [7a, b] the point of highest temperature of the distillation curve undertaken in the API-Daubert procedure is that of 95 % vol. (whereas current edition of API book [2] and [3] temperature for 100 % vol. is considered, like in present work).

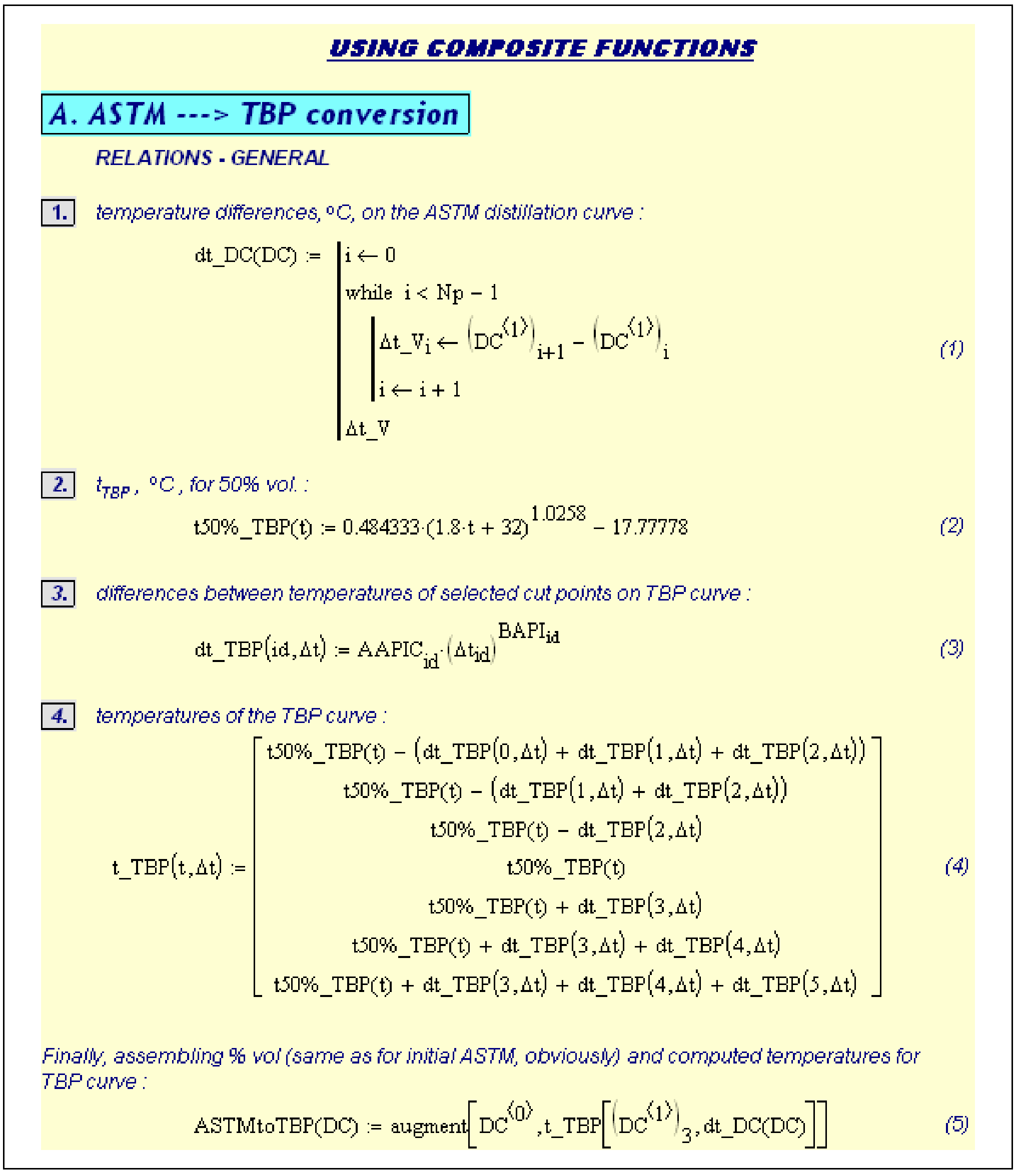
In step 1 (Fig. 1) temperature differences between temperatures of to 0, 10, 30,… % vol. distillate are computed as values of function "dt_DC", which has as argument (matrix) "DC". Function "dt_DC" is defined using a programming module written with programming operators (bold vertical lines). Note that programming modules can be indeed used for constructs other than definitions of functions. An equivalent definition, with a detailed and explicit writing of temperature differences and leading to same vector result, but not using programming, have been given in [1b].

Even if commands in programming module of Fig. 1 are almost self explanatory, some short explanations are following.

Index is initialized in the first line with the value 0, since according default settings numbering of the elements of a vector starts with 0 (not with 1). Within program modules of Mathcad assignments must be made with a left arrow. We emphasize that left-arrow definitions are local to the programming module and have no effect outside the module. As a general rule of thumb, while all assignments made in the worksheet (before and) outside the programming module are automatically 'passed' inside the module, the opposite is false, except the content of the last line of the module (see also next).

A 'while' loop begins in the second line of the programming module. All temperature differences are calculated consecutively inside the loop. As detailed in [1a], operator "$^{<n>}$" (written as exponent at matrix name) picks up the (n-1) th column of the matrix.

Since "DC" is the matrix with the distillation curve in the format described previously, " $DC^{<1>}$ " is leading to a vector containing the temperatures of selected points on the distillation curve. All Nd - 1 (= 6, six) temperature differences are thus computed one by one in the third line. In the fourth line, counter (index) "i" is increased (sequentially) by unity within every iteration, until it reaches the value Np - 1, that is 5. Temperature differences thus computed are stored in vector "Δt_V".



USING COMPOSITE FUNCTIONS

A. ASTM ---> TBP conversion

RELATIONS - GENERAL

1. *temperature differences, °C, on the ASTM distillation curve :*

```
dt_DC(DC) := | i ← 0
             | while  i < Np − 1
             |     | $\Delta t\_V_i \leftarrow \left(DC^{\langle 1\rangle}\right)_{i+1} - \left(DC^{\langle 1\rangle}\right)_i$
             |     | i ← i + 1
             | Δt_V
```
(1)

2. *t_{TBP}, °C, for 50% vol. :*

$$t50\%\_TBP(t) := 0.484333\cdot(1.8\cdot t + 32)^{1.0258} - 17.77778 \qquad (2)$$

3. *differences between temperatures of selected cut points on TBP curve :*

$$dt\_TBP(id,\Delta t) := AAPIC_{id}\cdot\left(\Delta t_{id}\right)^{BAPI_{id}} \qquad (3)$$

4. *temperatures of the TBP curve :*

$$t\_TBP(t,\Delta t) := \begin{bmatrix} t50\%\_TBP(t) - (dt\_TBP(0,\Delta t) + dt\_TBP(1,\Delta t) + dt\_TBP(2,\Delta t)) \\ t50\%\_TBP(t) - (dt\_TBP(1,\Delta t) + dt\_TBP(2,\Delta t)) \\ t50\%\_TBP(t) - dt\_TBP(2,\Delta t) \\ t50\%\_TBP(t) \\ t50\%\_TBP(t) + dt\_TBP(3,\Delta t) \\ t50\%\_TBP(t) + dt\_TBP(3,\Delta t) + dt\_TBP(4,\Delta t) \\ t50\%\_TBP(t) + dt\_TBP(3,\Delta t) + dt\_TBP(4,\Delta t) + dt\_TBP(5,\Delta t) \end{bmatrix} \qquad (4)$$

Finally, assembling % vol (same as for initial ASTM, obviously) and computed temperatures for TBP curve :

$$ASTMtoTBP(DC) := augment\left[DC^{\langle 0\rangle}, t\_TBP\left[\left(DC^{\langle 1\rangle}\right)_3, dt\_DC(DC)\right]\right] \qquad (5)$$

Fig. 1. Part of the Mathcad worksheet with definitions of functions needed for estimation of TBP curve from ASTM distill curve according API procedure [2]

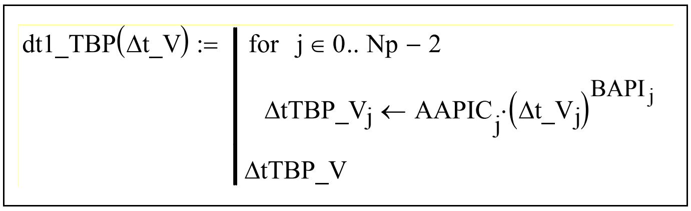
The value returned by the program is assigned to the variable placed on the last line and that is the reason why vector " Δt_V " appears in the last line of the programming module of relation (1) (Fig. 1). Further details on programming may be found in Mathcad help and in the Resource Center [5], well documented with explanations and numerous examples on this topic.

A 'for' loop may indeed be used in the programming module instead the "while" loop. Use of such 'for' loop is exemplified next, with the definition of function " dt1_TBP " (relation (a) next; in this case recursive increase of index "i" by unity is carried out implicitly by the "for" command).

In step 2 the temperature (℃) for 50% vol. on TBP curve is computed based on the temperature for 50% on the ASTM distillation curve. The relation between the two temperatures at 50% is same with that already given in [1a], and is written here as the function " t50%_TBP(t) ". Again, mnemonic notations are used in relation (2) for both function name and generic "t" for argument (temperature) (℃).

In step 3 differences between the temperatures of selected points on TBP curve are computed using the six correlations documented in Part 1 [1a]. These correlations are written in the Mathcad spreadsheet (Fig. 1) as a single function with two arguments, one being the range variable "id" (same as in step 1) and the other temperature differences (on ASTM curve). Comments on this kind of definition are provided in the second part [1b].

Alternative definitions for the function in relation (3) are possible, for example the next one in relation (a), which is using a programming module with a 'for' loop and is yielding same result.



```
$dt1\_TBP(\Delta t\_V) :=$  for  $j \in 0..Np-2$
                              $\Delta tTBP\_V_j \leftarrow AAPIC_j \cdot (\Delta t\_V_j)^{BAPI_j}$
                            $\Delta tTBP\_V$
```

(a)

In relation (a) "j" is a generic notation for index, which value is to be previously mentioned in the worksheet, and standing for "id". However, when using the suite of definitions as it will be described next, there is no more necessary to assign explicitly or to specify in the worksheet particular numerical values for "j".

Temperatures of the TBP distillation curve are then determined in step 4 by correspondingly subtracting or adding the appropriate temperature gaps (found in previous step) from t50% of TBP curve (found in step 2), using the two-argument function " t_TBP ". This function can be looked at as a composed function where composition is made by directly substituting in the written definition : a) the temperature for 50% on the TBP curve by the value of function " t50%_TBP " (which has been defined in equation (2), step 2) ; b) the temperature differences of TBP curve by values of function " dt_TBP " (already defined in equation (3), step 3). Hence, notation "t" for the first argument is corresponding to the temperature at 50% vol. on ASTM curve (needed as input in function " t50%_TBP "), "Δt" for temperature intervals on the same ASTM curve (required as input – second variable – in function " dt_TBP ").

We underline thus that in a definition like that used for " t_TBP " in step 4 (relation (4)), the two arguments, "t" and "Δt", are thus actually temperatures and temperature differences, respectively, belonging to initial (ASTM) curve.

The definition of " t_TBP " in (4) might be also written with programming module containing recursions like in (1) of Fig. 1 or in relation (a). But such a definition would have been tangled and the loss of readability wouldn't worth such simple calculus like the few additions or subtractions needed in the fourth step of the procedure.

In relation (5), 'augment' command in the definition of function " ASTMtoTBP " is used for joining side by side a vector with the volume percentages (first argument) and a vector with the computed temp of ASTM curve (second argument), in order to finally retrieve the desired TBP curve under the usual format adopted for the representation of distillation curves, that is a 7×2 matrix.

In the same relation (5), TBP temperatures are calculated with two-argument function " t_TBP ", which is composed (through its second argument) with function " dt_DC " previously defined and meant to calculate the temperature differences based on the initial ASTM curve (" DC "). Hence, definition of " ASTMtoTBP " like in (5) uses a series of two function compositions, " augment∘t_TBP∘dt_DC ", coincidently through the second argument of both functions 'augment' and " t_TBP ".

Even if now obvious, we point out that the last function, “ ASTMtoTBP ”, has a sole argument, the matrix with data of the initial (ASTM) distillation curve, and the desired TBP curve as output.

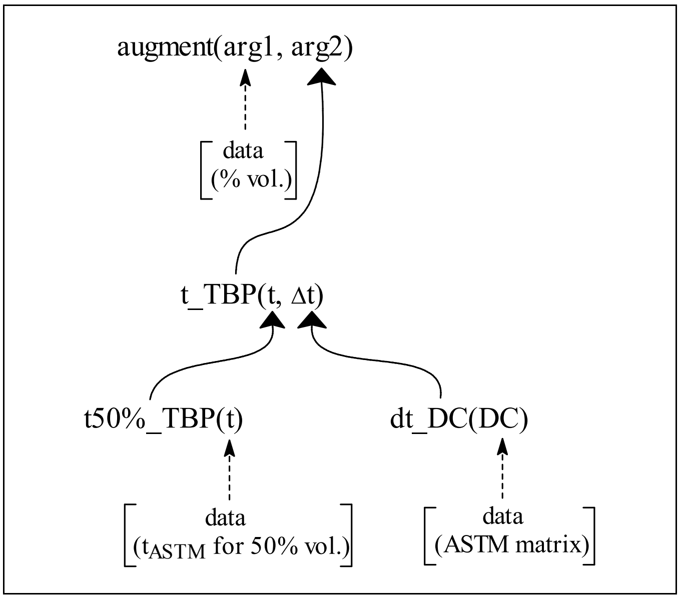


Fig. 2. Representation of the composed functions employed in Mathcad worksheet of Fig. 1. Curved (continuous line) arrows represent compositions of functions and dashed-line arrows show direct input with given data (from known ASTM distillation curve)

Since its definition takes into account all definitions and assignments made previously in the Mathcad worksheet, function “ ASTMtoTBP ” allows to directly compute within a single call the TBP curve from a given ASTM curve (see also next the example in Fig. 3).

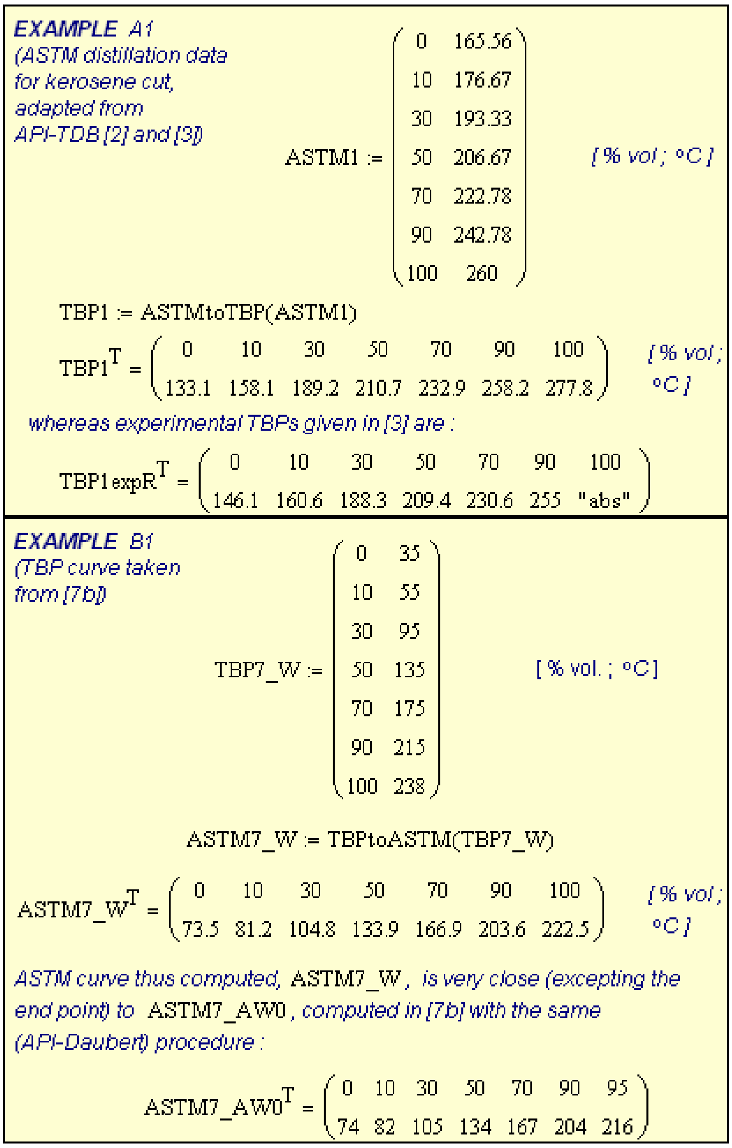
Compositions of various functions, as previously described, is illustrated in Fig. 2.

Use of operators such as “tree” operator (as below in expression (b)), or infix or similar operators available in Mathcad, may render more legible the writing of composition of functions.

$$\text{ASTMtoTBP1(DC)} := \begin{array}{c} \text{augment} \\ \text{DC}^{\langle 0\rangle} \qquad\qquad \text{t\_TBP}\left[\left(\text{DC}^{\langle 1\rangle}\right)_3, \text{dt\_DC(DC)}\right] \end{array} \qquad \text{(b)}$$

An example with the conversion of an ASTM distillation curve into the TBP curve according API method is shown in Fig. 3 (example A1, ASTM distillation data taken from [2]). Employing previous functions definitions, it is now possible to compute the TBP curve by a single call of function “ ASTMtoTBP ” and merely replacing the generic argument “DC” with the given ASTM distillation data (matrix).

TBP curve thus determined is obviously identical with that computed with the same procedure in the original API book [2] or in [3], from which the example is borrowed. (Temperature for 0% was taken from the latter reference, whereas former reference does not provide temperatures at 0% neither for 100% vol.; temperature for 100 % is an extrapolated one, since neither book of the two gives its value).



EXAMPLE A1
(ASTM distillation data for kerosene cut, adapted from API-TDB [2] and [3])

$$\mathrm{ASTM1} := \begin{pmatrix} 0 & 165.56 \\ 10 & 176.67 \\ 30 & 193.33 \\ 50 & 206.67 \\ 70 & 222.78 \\ 90 & 242.78 \\ 100 & 260 \end{pmatrix}$$ *[% vol ; °C]*

TBP1 := ASTMtoTBP(ASTM1)

$$\mathrm{TBP1}^T = \begin{pmatrix} 0 & 10 & 30 & 50 & 70 & 90 & 100 \\ 133.1 & 158.1 & 189.2 & 210.7 & 232.9 & 258.2 & 277.8 \end{pmatrix}$$ *[% vol ; °C]*

whereas experimental TBPs given in [3] are :

$$\mathrm{TBP1expR}^T = \begin{pmatrix} 0 & 10 & 30 & 50 & 70 & 90 & 100 \\ 146.1 & 160.6 & 188.3 & 209.4 & 230.6 & 255 & \text{"abs"} \end{pmatrix}$$

EXAMPLE B1
(TBP curve taken from [7b])

$$\mathrm{TBP7\_W} := \begin{pmatrix} 0 & 35 \\ 10 & 55 \\ 30 & 95 \\ 50 & 135 \\ 70 & 175 \\ 90 & 215 \\ 100 & 238 \end{pmatrix}$$ [% vol. ; °C]

ASTM7_W := TBPtoASTM(TBP7_W)

$$\mathrm{ASTM7\_W}^T = \begin{pmatrix} 0 & 10 & 30 & 50 & 70 & 90 & 100 \\ 73.5 & 81.2 & 104.8 & 133.9 & 166.9 & 203.6 & 222.5 \end{pmatrix}$$ *[% vol ; °C]*

ASTM curve thus computed, ASTM7_W *, is very close (excepting the end point) to* ASTM7_AW0 *, computed in [7b] with the same (API-Daubert) procedure :*

$$\mathrm{ASTM7\_AW0}^T = \begin{pmatrix} 0 & 10 & 30 & 50 & 70 & 90 & 95 \\ 74 & 82 & 105 & 134 & 167 & 204 & 216 \end{pmatrix}$$

Fig. 3. Part of Mathcad worksheet with example of distillation curves conversion: A1 – from ASTM curve into TBP curve using the composed function defined in Fig. 1 ; B1 – from TBP curve into ASTM distillation curve (In example A1, temperature for 0% taken from [3] and temperature for 100% estimated by present author. Example B1 is placed in the Mathcad worksheet after definition (9) of Fig. 4; an extrapolated value is assigned to t100% of the above TBP curve, “ TBP7_W ”).

Conversion of TBP curve to ASTM distillation curve

The API method may be applied backward, to estimate the ASTM distillation curve starting from a (known) TBP curve. The relations needed for this reversed procedure can be readily derived from those used for the direct method and they have already been given in [1a]. Mathcad version of API procedure applied for reversed transformation, that is from TBP curve to ASTM distillation curve, is shown in the part of Mathcad worksheet given in Fig. 4. Both this part of worksheet and that given in Fig. 1, as well as the examples (Fig. 3), have actually been written in the same Mathcad worksheet (file).

In the first step, temperature intervals (*°C*) on the TBP curve are determined using same function “ dt_DC ” as previously defined in the worksheet (Fig. 1).

Temperature (*°C*) at 50% vol. on ASTM distillation curve is then computed in second step, based on temperature (*°C*) for 50% vol. on TBP curve and using function “ t50%_ASTM ”, which is defined according relation (6) (Fig. 4).

In third step temperature gaps on ASTM curve are computed with function “ dt_ASTM ” defined by relation (7), where function “ dt_ASTM ” is analogous to “ dt_TBP ” used in the direct method and defined by (3). The six (Nd = 6) correlations represented by the two-argument function “ dt_ASTM ” have been conspicuously derived by rearrangement of (3) in order to get the explicit expressions for ASTM temperature differences.

In the fourth step, temperatures of the ASTM curve are determined with function “ t_ASTM ”, which aim is to subtract or to add the temperature gaps found in step 3 to beforehand determined t50% of ASTM distillation.

Instead of the two functions “ t_TBP ” and “ t_ASTM ” (definition (4) in Fig. 1, and (8) in Fig. 4), use of a single common function for both fourth steps of direct and reversed procedures is also possible, since the two functions are performing analogous jobs (similarly to common use of “ dt_DC ” in first steps). A definition of such function of joint usage would have been again tedious and of reduced readability and wouldn’t have been worth the simple calculations. Moreover, being given the mnemonic notations employed, it appeared much more convenient to write and employ two distinct functions, “ t_TBP ” and “ t_ASTM ” to be specific, in our Mathcad worksheet (Figures 1, 4).

Finally, function “ TBPtoASTM ”, which definition in (9) includes “augment” command, is meant to retrieve the estimated ASTM curve in the consecrated format, that is a 7×2 matrix.

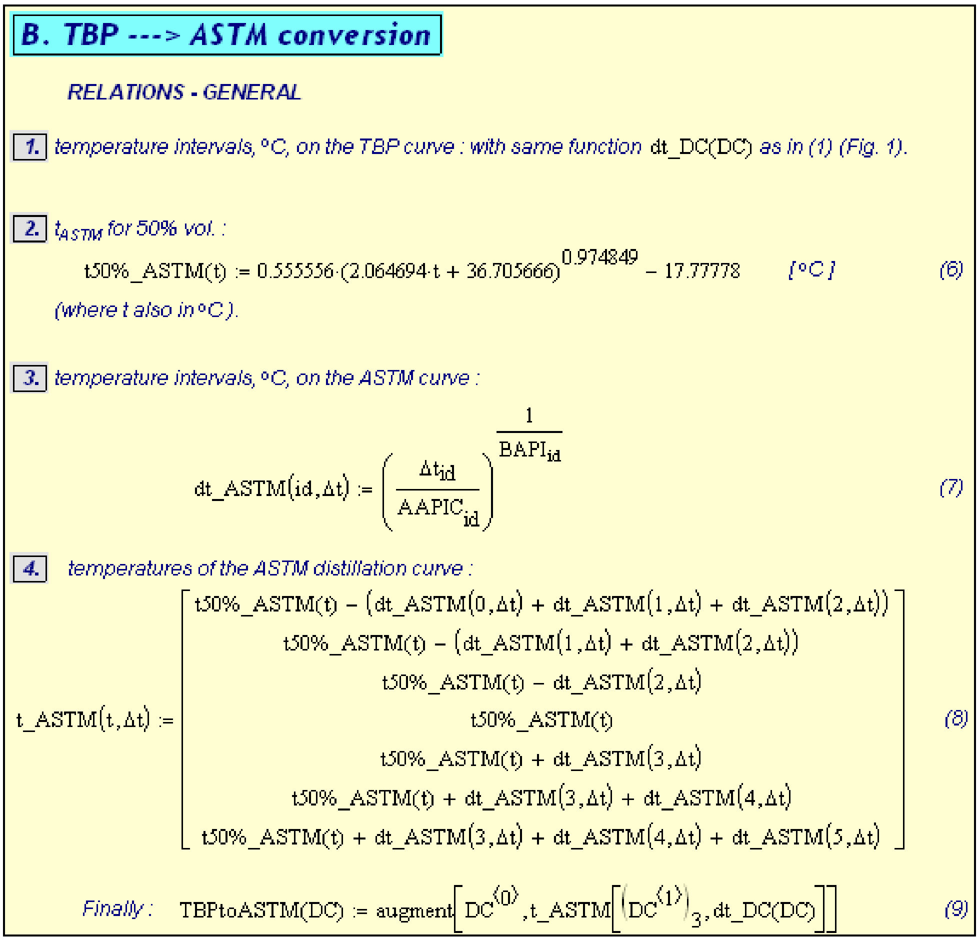
Compositions of functions used within definitions of “ t_ASTM ” and “ TBPtoASTM ” are analogous with that previously described for “ t_TBP ” and “ ASTMtoTBP “, respectively. It is worth to remind that the two arguments, “t” and “Δt”, of the function “ t_ASTM ” as defined by (8), are belonging to the initial (given) distillation curve, similarly to arguments with same notations in the definition of “ t_TBP ” in (4) for the direct method.

Example B1 in Fig. 3 illustrates the simple utilization of the procedure for the conversion of TBP curve into ASTM distillation curve, with a mere call of the (composed) function “ TBPtoASTM ” and replacement of generic (independent) variable “DC” with the matrix of the given TBP curve.

Both examples in Fig. 3 prove how convenient it is the use of the API procedures, once these having been transposed into Mathcad as previously described.

When employing the herein given Mathcad version of API procedure, definitions of parameters AAPI, BAPI and of the series of functions (1)-(5) and (6)-(9) may be written in a separate Mathcad worksheet (file) which can be hitherto included in the current worksheet with the “Reference” command.

Alternatively, the mentioned definitions may be written in areas inserted within the current worksheet, areas which can eventually be subsequently collapsed (becoming thus hidden but still calculating). These two features of Mathcad environment, the “Reference” command and the inserted (and collapsed) area, has been exemplified in previous part [1b] (for details see also corresponding items in Help and in Resource Center of Mathcad [5]).



B. TBP ---> ASTM conversion

RELATIONS - GENERAL

1. *temperature intervals, °C, on the TBP curve : with same function* dt_DC(DC) *as in (1) (Fig. 1).*

2. t_{ASTM} *for 50% vol. :*

$$t50\%\_ASTM(t) := 0.555556\cdot(2.064694\cdot t + 36.705666)^{0.974849} - 17.77778 \qquad [°C] \qquad (6)$$

(where t also in °C).

3. *temperature intervals, °C, on the ASTM curve :*

$$dt\_ASTM(id,\Delta t) := \left(\frac{\Delta t_{id}}{AAPIC_{id}}\right)^{\frac{1}{BAPI_{id}}} \qquad (7)$$

4. *temperatures of the ASTM distillation curve :*

$$t\_ASTM(t,\Delta t) := \begin{bmatrix} t50\%\_ASTM(t) - (dt\_ASTM(0,\Delta t) + dt\_ASTM(1,\Delta t) + dt\_ASTM(2,\Delta t)) \\ t50\%\_ASTM(t) - (dt\_ASTM(1,\Delta t) + dt\_ASTM(2,\Delta t)) \\ t50\%\_ASTM(t) - dt\_ASTM(2,\Delta t) \\ t50\%\_ASTM(t) \\ t50\%\_ASTM(t) + dt\_ASTM(3,\Delta t) \\ t50\%\_ASTM(t) + dt\_ASTM(3,\Delta t) + dt\_ASTM(4,\Delta t) \\ t50\%\_ASTM(t) + dt\_ASTM(3,\Delta t) + dt\_ASTM(4,\Delta t) + dt\_ASTM(5,\Delta t) \end{bmatrix} \qquad (8)$$

Finally :

$$TBPtoASTM(DC) := augment\left[DC^{\langle 0\rangle}, t\_ASTM\left[\left(DC^{\langle 1\rangle}\right)_3, dt\_DC(DC)\right]\right] \qquad (9)$$

Fig. 4. Part of the Mathcad worksheet with definitions of functions needed for estimation of ASTM distillation curve from TBP curve according to API procedure [2]

Conclusion

An easy-to-use Mathcad implementation of API procedure for ASTM – TBP distillation curves interconversion is provided. The procedure transposed into Mathcad can be used with a single call of a one-variable function, offering thus a simple and compact utilization. The mentioned function is defined as the result of several compositions of functions. Moreover, Mathcad-written procedures for direct or backward conversion between the two types of distillation curves can be used for repeated or iterative tasks.

Tools of Mathcad environment, such as programming module, “tree” operator, operators for vector or matrix manipulation like “augment”, have been employed. Other Mathcad features, such as vector-valued functions or “Reference” command - to mention a few among most relevant -, have also been used, like in [1b] with the previous Mathcad version of the API procedure.

Notations and Acronyms

AAPI – vector, with elements the values of parameters of the API procedure (for converting temperature differences)

arg – argument (of function)

ASTM – distillation curve, according ASTM D 86 (as part of functions or matrices names)
BAPI – vector, with elements the values of parameters of the API procedure (for converting temperature differences)
DC – matrix with distillation curve (given under the format described in text ; also as part of functions names)
dt – difference between temperatures (in functions names)
id – index (for counting temperature differences, id = 0, 1, …, Nd – 1)
j – index (for counting temperature differences within 'for' loop, j = 0, 1, …, Nd – 1)
Nd – number of temperature differences (Nd = 6)
Np – number of selected points on the distillation curve (Np = 7)
t – temperature, °*C* (also in functions names)
TBP – True Boiling Point (as part of functions or matrices names)
V – vector (in vectors names)
Δt – difference between temperatures (corresponding to consecutive selected points on distillation curve (temperature intervals), °*C*.

References

1. Filotti, L. - a) *Buletin UPG, Seria Tehnică*, **LXII**, No. 1, 2010, pp. 45-54 (Part 1) ; b) *Buletin UPG, Seria Tehnică*, **LXII**, No. 3A, 2010, pp. 169-179 (Part 2);
2. Daubert, Th. E. - in *Tehnical Data Book*, American Petroleum Institute, 7th ed., 2005, pp. 3-6 (Procedure 3A1.1, in Ch. 3, „Petroleum Fraction Distillation Interconversions", same as in 6th ed., 1997);
3. Riazi, M. R. - *Characterization and Properties of Petroleum Fractions*, ASTM-International, West Conshohocken (PA), 2005, pp. 103-105;
4. Daubert, T. E. - Petroleum Fraction Distillation Interconversion, *Hydrocarbon Processing*, **73**, No. 9, 1994, pp. 75-78.
5. * * * - a) Mathcad Help ; b) Mathcad Resource Center.
6. Montemayor, R. G.- in *Distillation Vapor Pressure Measurement Petroleum Products*, (Montemayor, R. G., Ed.), ASTM International, West Conshohocken (PA), 2008, p. 6;
7. Paradowski, H. - in *Petroleum Refining* (Wauquier, J. P., Ed.), vol. 1, Technip – Institut Français du Pétrole, Paris, 1995, a) p. 101; b) p. 165.

Transpunere în Mathcad uşor de utilizat pe calculator a metodelor API de interconversie a curbelor de distilare ale fracțiunilor petroliere. Partea a 3-a. Proceduri de interconversie a curbelor SR ISO şi PRF (Puncte Reale de Fierbere) scrise în Mathcad folosind funcții compuse

Rezumat

Procedurile recomandate de API (American Petroleum Institute [2]) pentru interconversia curbelor de distilare SRO (cf. SR ISO 3405:1998, în trecut „STAS", cf. STAS 36-1980 ; echivalent cu ASTM D 86) şi PRF (Puncte Reale de Fierbere, aproximativ cf. STAS 2150-1980) au fost transpuse în platforma Mathcad. Versiunile Mathcad prezentate în această parte a lucrării, atât pentru trecerea de la curba ASTM la curba PRF cât şi pentru transformarea inversă, de la curba PRF la curba de distilare ASTM, sunt mai concise, mai uşor de utilizat şi de o aplicabilitate mai generală decât versiunile date în precedentele părți [1a, b]. În prezenta versiune relațiile folosite de către procedura API sunt scrise în Mathcad sub formă de funcții sau de funcții compuse, într-o secvență ce în final permite utilizarea procedurii într-o unică apelare a unei funcții de o singură variabilă. Folosirea procedurilor transpuse în Mathcad este ilustrată cu exemple.