Chapter 2
Working with Physical Quantities: Problems and Solutions

Valery Ochkov and Konstantin Orlov

Abstract This chapter deals with the use of dimensional quantities in thermal calculations. The main aspects of work with dimensional variables in thermal calculations are given. The concepts of physical and empirical formulas are extended by pseudo-empirical formula concept. Few samples are provided.

Programming languages and spreadsheet programs usually calculate with simple quantities (numbers) rather than with physical quantities, like mass, velocity, energy, etc. That reduces the "readability" of the calculations, slows them down, and is additionally a source of errors [8].

The consequences of misusing units can be extremely significant and even cause real disaster. Here are three examples, of which the first two directly related to heat-engineering.

A Boeing 767 belonging to Canadian airlines was flying on 23 July 1983 from Montreal to Edmonton. Halfway (at an altitude of 12 500 m) the plane was forced to make an emergency landing at an abandoned military airfield in the town of Gimli, because of empty fuel tanks. The reason for the emergency was a simple calculation error—the mass of fuel needed for the trip was incorrectly calculated. Usually Air Canada used the British metric system (pounds) for the calculation of the needed fuel. However, this airplane was a new one, where the SI-units had to be used (kg). In order to calculate the needed mass of fuel the volume had to be multiplied with

The site of the chapter: https://www.ptcusercommunity.com/message/423012.

V. Ochkov (✉) · K. Orlov
Moscow Power Engineering Institute, Moscow, Russia
e-mail: ochkov@twt.mpei.ac.ru

K. Orlov
e-mail: orlov@twt.mpei.ac.ru

V. Ochkov · K. Orlov
Joint Institute for High Temperatures, Russian Academy of Sciences, Moscow, Russia

V. Ochkov et al., *Thermal Engineering Studies with Excel, Mathcad and Internet*,
DOI 10.1007/978-3-319-26674-9_2

the density. However, the density was used with wrong units and therefore the tanks have been just half-full.

A similar mistake has led to the loss of the American satellite Mars Climate Orbiter (MCO). MCO launched on 11 December 1998 with a PH-2 Delta rocket. The unit arrived at Mars after 9 months, at the 23 September 1999. The engine should break the unit and go into the orbit of Mars. After 5 min, when the satellite went behind Mars, no more signals have been detected. From the analysis of data the scientists found out, that the device has passed the surface at a height of 57 km instead of 140 km and simply burned in the atmosphere (unit cost $125 million). This large deviation was caused by the fault of experts, who prepared the mission: when calculating the braking pulse one of the groups used Imperial-units (pound-force) and the other SI-units (newton).

You can mention another engineering error associated not with units themselves, but with the scale of measurement. In the past was a bridge across the Rhine built. On one side the Germans started and on the other side the Swiss started to build it. When both side met, they recognized that the height difference is about a half meter. The reason for the error was that the German standard of the construction zero height is the average level of the North Sea, and in Switzerland the average level of the Mediterranean Sea.

Any expert, who works in the field of science and technology, can tell numerous examples of such errors and incidents associated with the incorrect handling of units. The transition from calculations by hand to "dimensionless" programming languages, could not either solve this problem.

Mathcad—it's not just physical/mathematical program. Often people do not use units and just mention the unit of the used constant/variable just in the comments.

So you can, for example, often see that the pressure is just written down in Mathcad as: **p := 120**, instead of using the more clear and correct way: **p := 120 atm**.

What are the reasons for the under-utilization of Mathcad? Firstly, some users are just unaware of this useful tool. It is possible to use units in Mathcad and then nevertheless transfer the calculation to another programming language or spreadsheet program, in which, I repeat, the variables are stored as numerical values and their units noted in the comments.

There is a second group of Mathcad-users, which do not use physical quantities in their calculations, explaining or justifying it by the fact that all quantities prescribed in only one fundamental measurement system (e.g., international SI-Units) and they do not have any problems with the translation of the units. This motivation often buttressed by the fact that Mathcad-document without physical quantities is much easier to prepare for transferring to other programming languages. In addition, a Mathcad-document with disabled mechanism to work with units of measurements works faster.

The third and the main cause of the failure of many users to work physical quantities in the calculations lies deeper. It is associated with some features and shortcomings associated with the work with physical quantities, which even the most experienced users sometimes force to translate them into the category of comments.

However, using comments represents extra work. Often the user says to himself that he will add the necessary comments later. Often this “later” doesn’t come and the comments are never added. Nevertheless, when someone wants to use this in the past-created document he cannot understand which values he has to use for the variables, because he does not know which units are used there. Even for the creator of the document it can be difficult to remember which units are need to be used, because the document was created too long ago in the past.

2.1 Tools for Using Physical Quantities in Mathcad

Using units in Mathcad is quite simple. When you enter a numeric value you can connect it with units through typing in a built-in/user-unit or also a functional dependence. Additionally you can also choose a unit through the dialog box (Fig. 2.1). Usually the multiplication sign between the numerical value and unit is invisible. When you enter a dimensional value the option can be given to the user to select the unit of measurement, for example, through the use of switches: the temperature/pressure can be shown at different temperature or pressure scales with different units of temperature/pressure.

Among Mathcad (Ver. 13, 14, 15) it became possible not only to work with the built-in units (SI, MKS (meter, kilogram, second), CGS (centimeter, gram, second) and the U.S. Units), but also with custom units. This in particular means that the user can modify one of the predefined units or create even new units. Further calculations with custom units are in the same way monitored like calculations with old units: meters, for example, cannot be added with kilograms; which helps to avoid errors and typos.

When you use the “ = ” operator Mathcad will display the following: ∎ = ∎ ∎. The first place holder contains variable name, the second one the numerical value, and the third one the unit. By clicking on the third placeholder you can change the used unit and this convert to another system of measurement (e.g., bar → Psi). In some cases it is useful to duplicate the result value and show this way the result with different units; so the reader can choose, which one is the most convenient one for him.

What forces even experienced users to avoid units in purely physical calculation or to withdraw them from an almost finished document?

These “pitfalls”:

Some Mathcad tools are not designed to work with dimensional values. They interrupt the work or return an error message such as “There should be no dimension value” or, much even worse, give the wrong answer. In such cases it is necessary to temporarily withdraw the units by dividing the variables by the unit, and then after the calculation add the unit by multiplying the variable with the base

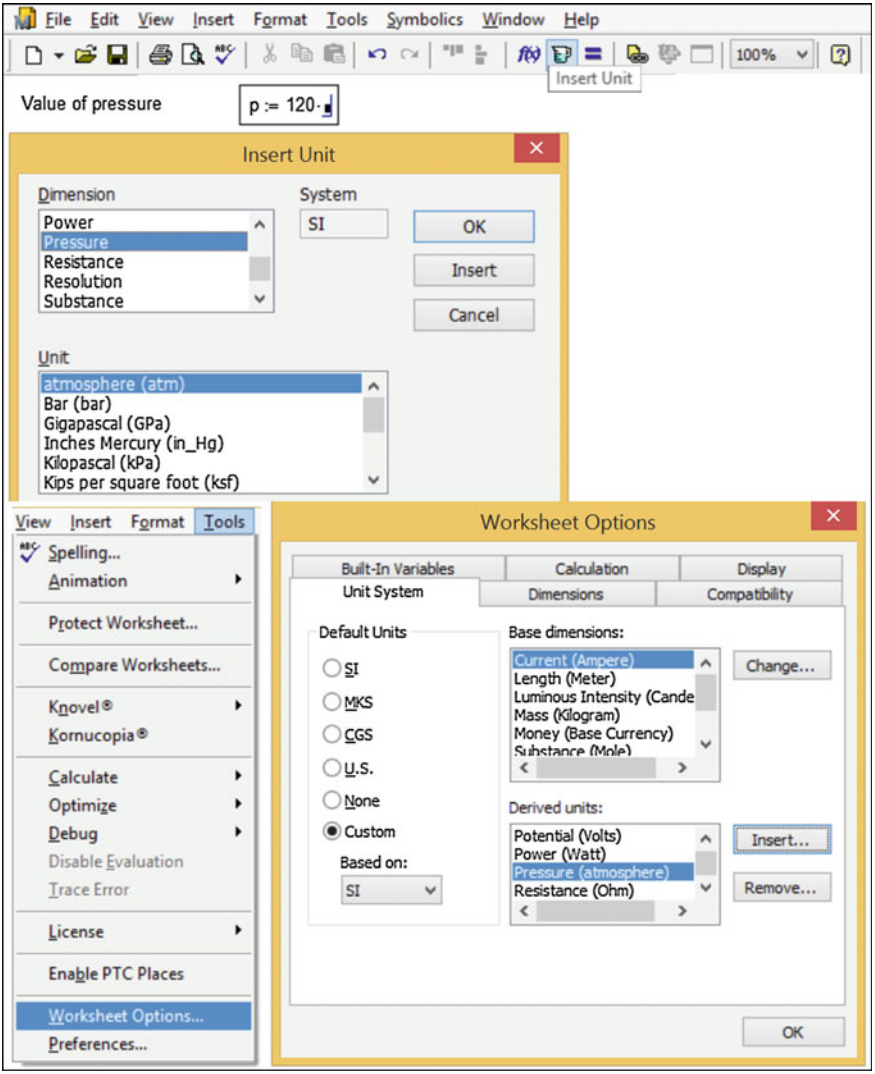

Fig. 2.1 Using units in Mathcad 15

unit. This method will be described in more detail when we talk in more detail about working with empirical formulas.

If the user does not plan to use physical quantities, then the use of units should be disabled. Therefor you have to make the following commands: press Tools, Options, and then a new window will pop up. There you press on the button Unit System and set the switch to No (see Fig. 2.1). This way it is possible to avoid errors. For example, when the user wants to create a function like f(x) = x^2, and accidently uses instead of x an unit like: **A**, **S**, **V**, etc. Mathcad will not return an error as expected. When using Quickplot we will see just one point instead of the functional correlation. When turning of units we can use these former unit-names also as variables (like x). This way mistakes can be avoided.

Symbolic math units operate in Mathcad as simple variables, they do not take into account that 1 m is equal to 100 cm or an hour has 60 min, etc. The symbolic math tool is a foreign element in Mathcad. It is taken out of the program Maple, in which the units have appeared only in the eighth version (Mathcad 5–13), or out of the program MuPAD, in which are either no units of measurement used (Mathcad 14/15 and Mathcad Prime). Symbolic Mathematics is an auxiliary tool in Mathcad, which is rarely used in calculations, and only plays a secondary role. If you want to use units in symbolic mathematics, than more should be done: it is necessary to use the substitute command in Mathcad, and tell Mathcad that 1 m = 100 cm, 1 h = 60 min, etc.

In Mathcad 15 you can store only dimensionless quantities or quantities of the same dimension in vectors/matrices, i.e., the same physical quantity like time, strength, weight, etc. There is known only one exception to this rule. The function Find, for example, can return a vector with mixed elements. If you need the results of several different function united in a vector or matrix, it is possible to deprive the units first and then bring them back. This restriction has been removed in Mathcad Prime.

Often so-called empirical formulas are used—formulas that relate not just on physical interrelations. Example of an empirical formula showed on Fig. 2.2 provides the information how to calculate with Mathcad the (estimated) used car value depended on age and mileage.

The empirical formula, with which the price of an old car can be calculated, was created with the help of statistical processing. This example shows how to supplement these formulas without interference with physical quantities. When empirical formulas are published it is always clearly marked, what units should be used as the initial values and in what units the answer will be. In this case, the vehicle age must be entered in years (yr) and mileage in Miles (mi). The formula returns the car price in U.S. dollars ($). These variables must lose their units in the formula. Figure 2.2 shows the formula of the **CarPrice** where variables are divided by their assigned units, the formula itself is multiplied by the currency. (Figure 2.2 was created at a time when the U.S. dollar was worth 50 rubles.)

The list of usable units in Mathcad does not contain all units. There are missing, for example, often used units in information technology, like bits, bytes, etc. In Mathcad 14 are also missing currency-units (dollar, ruble, euro, etc.). Mathcad 14 deals only with “seven” SI-Units (time, length, mass, current, temperature, light intensity and amount of substance) and their derivatives—units, which composed of these basic units (force, energy, power, etc.). And how to deal with older Mathcad versions, if you want to perform the so-called economical calculations, in which are dollars, rubles, euros, etc. are present? One solution is shown in the upper part of Fig. 2.2. The Dollar ($US) is assigned with a random unit of measurement, a physical quantity which does not appear in this calculation—candela (unit of luminous intensivity): **$:= cd** (see the first operator in Fig. 2.2). Next, you can

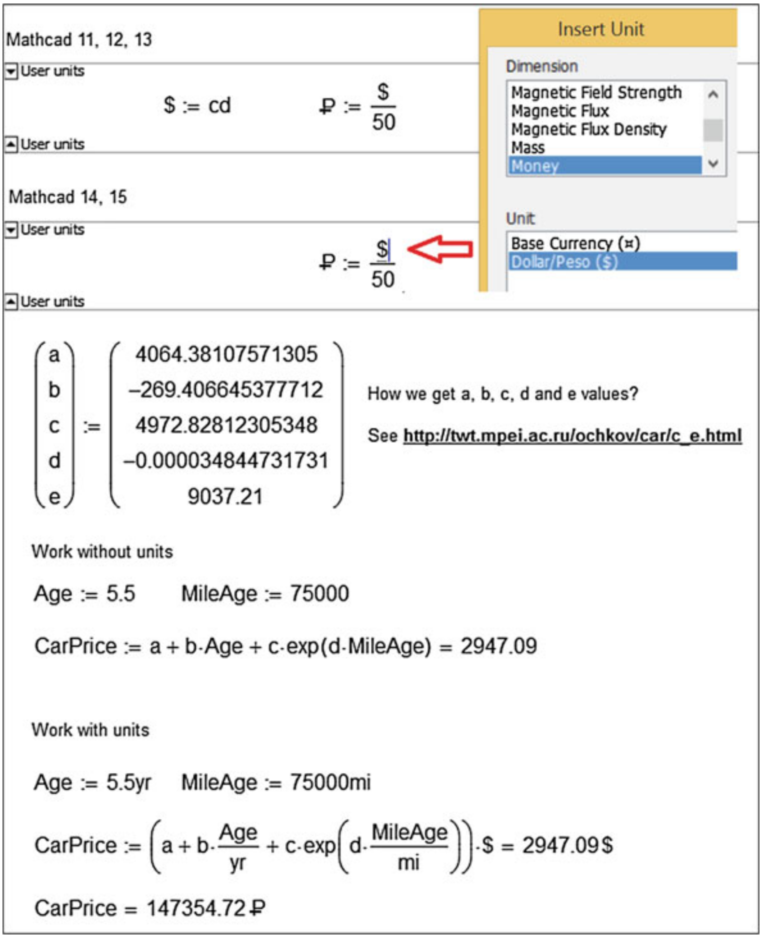

Fig. 2.2 Working with empirical formulas

connect the dollar with other currencies used in the calculation, for example, **Rub := $/50**. After that the price will be displayed by default, candelas (Mathcad 14 and 15), or in base currency units, which has to be replaced by the appropriate unit—dollars or rubles (see the last row in Fig. 2.2).

Another example shows Fig. 2.3, where we have to solve the following problem of thermal engineering: there is a power plant efficiency η given and you need to calculate the specific fuel consumption for electricity generation. All information for this calculation is given. The formula for the needed fuel is: **b := 12300/η**. It is said that the efficiency (**η**) must be expressed as a percentage (or in relative units) and the result of the needed fuel will tell us how many gram (**gm**) of fuel we need for the production of 1 kWh of electricity. For example: 12300/32 = 384.4 or

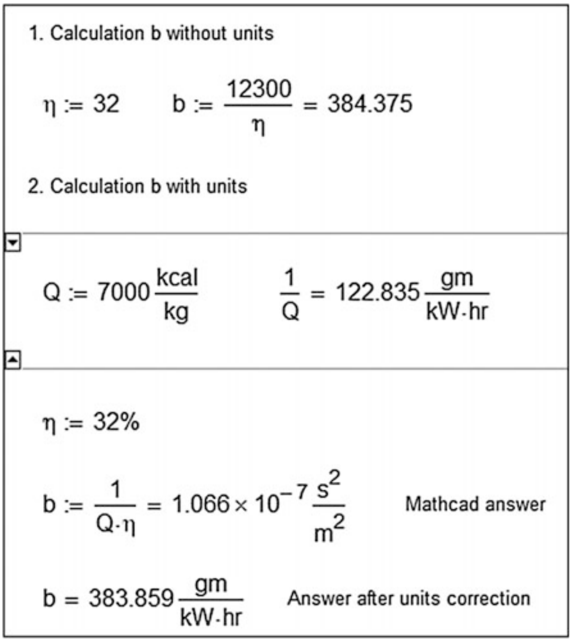

Fig. 2.3 Working with pseudo empirical formulas

123/0.32 = 384.4—thermal power plant with an efficiency of 32 % burns 384.4 g of conventional fuel for the generation of 1 kWh of electricity.

Like already often mentioned, it is possible to withdraw the units from all variables and then to obtain the answer dimensionless. The answer can either be multiplied afterwards with the demanded units or it is also possible to write the units just in the comment. Do not forget that these pseudo empirical formulas have been created with good intentions. This way we are liberated from additional unit conversions and these formulas are also very ease to remember/use. In this case (because of the easy task), it is also possible to obtain to answer by modifying the formula: **b := 1/(Q · η)**. Q is in this formula equal to the heating value, which has an assumed value of 7000 kcal/kg (the heat of combustion of coal with good quality). When you enter this modified formula you do not have to withdraw the units of the input and additionally you will get a more precise answer.

There formulas (a lot of them in the technical literature), which you cannot immediately understand and this can lead to errors in the calculation. For example, you need to calculate the molality of a solution out of its molarity. You go to the Internet, do some search and find the site with the desired formula (see Fig. 2.4).

This formula is easy to implement in Mathcad (see Fig. 2.5).

Figure 2.5 shows the encircled formula from table (see Fig. 2.4) implemented in Mathcad. The result of this formula looks very reasonable and this is also the crux, since nobody will check this formula for possible mistakes. All formulas in Fig. 2.4 are pseudo-empirical. Therefore you should check in which units all variable have to be entered. Otherwise Mathcad will convert the units how it is needed. And here http://erichware.info/sposob/ximia/ximikonc.htm

	0 < K < 100 % mass ratio	T titer	L molality	M molarity
M =	$\frac{10 * q * K}{wp}$	$\frac{1000 * T}{wp}$	$\frac{1000 * q * L}{1000 + wp * L}$	$\frac{1000 * yp}{v}$
L =	$\frac{1000 * K}{wp * (100 - K)}$	$\frac{1000 * T}{wp * (q - T)}$	$\frac{1000 * yp}{mb}$	$\frac{1000 * M}{1000 * q - wp * M}$
T =	$\frac{q * K}{100}$	$\frac{mp}{v}$	$\frac{q * wp * L}{1000 + wp * L}$	$\frac{wp * M}{1000}$
K =	$\frac{mp * 100\,\%}{mb + mp}$	$\frac{100 * T}{q}$	$\frac{100 * wp * L}{1000 + wp * L}$	$\frac{wp * M}{10 * q}$

wp – molar mass of solute (g/mole) mb – mass of solvent (g)

mp – mass of solute (g) yp – moles of solute (mole)

v – volume of solution (cm^3) q – density of solution (g/cm^3)

Fig. 2.4 Formula for the conversion of concentrations from the Internet

I want to calculate the molality (L) NaCl aqueous solution with a molarity M = 2 mole/L and the density of q = 1.076 gm/mL

$$M := 2\,\frac{mol}{L} \qquad q := 1.076\,\frac{gm}{cm^3} \qquad wp := 58.44\,\frac{gm}{mol}$$

$$L := \frac{1000\,M}{1000\,q - wp \cdot M} = 1.859\,\frac{mol}{kg}$$

Fig. 2.5 Conversion of Molarity → Molality

is the mistake. Although we use in this case even the right units we still get the wrong result. The Multiplier 1000 is integrated in the formula because of the relation between the units g/kg and cm^3/L. If we use the pseudo-empirical formula as a real empirical formula in Mathcad, we will get the correct result (see Fig. 2.6).

If we want to use the formula like a real physical formula we need to think about the multiplier 1000. In this case we need to delete it. After correcting the formula we will get the correct result (see Fig. 2.7).

$$L := \frac{1000\,\dfrac{M}{mol \cdot L^{-1}}}{1000\,\dfrac{q}{gm \cdot cm^{-3}} - \dfrac{wp}{gm \cdot mol^{-1}} \cdot \dfrac{M}{mol \cdot L^{-1}}} \cdot \frac{mol}{kg} = 2.085\,\frac{mol}{kg}$$

Fig. 2.6 Pseudo-empirical Formula as a purely empirical formula

$$L := \frac{M}{q - wp \cdot M} = 2.085\ \frac{mol}{kg}$$

Fig. 2.7 Using Formula like a physical one

Units support for physical quantities was till the 13th Mathcad version only possible on an absolute scale. This means in particular that if the value is zero itself, then it is not necessary to attribute any unit, although nevertheless it should be done: for example, **l := 0 m**, in order not to disrupt the possibilities to control the units. But often we also need a relative measurement scale for some tasks. For example, we usually measure the temperature in degrees Celsius (relative scale), and not in Kelvin (thermodynamic temperature—absolute scale). In Mathcad (also in new versions) we do not have a multiplication sign between the numeric value (**25**) and unit (**°C**), when we use a relative scale (**t := 25 °C**). We have more functional correlation between the unit and the numeric value, connected by a postfix-operator.

Figure 2.8 shows one possible solution of the problem of using relative measurement scales: given is the temperature t_1 (heater entrance) and the temperature difference **Δt** (difference between the entrance and exit). It is necessary to determine the temperature $\mathbf{t_2}$ at the outlet of the heater. There you can see three objects called **°C** (each time another User has to be selected in Mathcad): two constants named **°C**

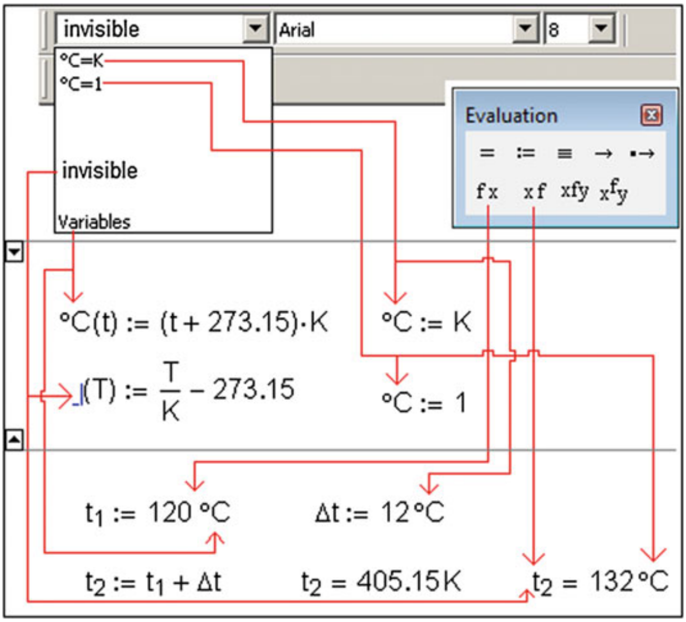

Fig. 2.8 Working with temperature scales

(the first is called **°C := K** and the second **°C := 1**) and a function called **°C**. Additionally you can see another function with an invisible name—**(T) := T/K-273.15**. Invisible function name can be entered by space sign (using Ctrl-Shift-K) or using text style with white color of characters.

The names of all three objects coincide (**°C**). However they are different objects, because they are used by different users in Mathcad (selectable via the button left next to the font-button). When working with relative temperature relative scales are three situations, in which the above introduced objects called **°C** are useful.

Situation 1

In the calculation it is necessary to enter the temperature in Celsius. To do this, we use the function called **°C** with the postfix operator. The input of the function will be temperature in **°C**. Internally Mathcad will convert the Celsius temperature into kelvin. Therefore all calculations can be carried in Mathcad, since Mathcad works with the absolute temperature scale.

Situation 2

You must enter the temperature difference **Δt**. It is very important that differences entered with another functions. Usuall (and very bad) mistake is to use °C. By using °C you will increase temperature difference value for 273.15 degrees. Since **Δ°C** is equal to **K** you can use the object **°C = K**.

Situation 3

You want to convert the temperature from Kelvin to Celsius scale. To do this you use the prefix operator and the invisible operator. The result will be first displayed without any units. So u have to use additionally the **°C = 1** in order to add some "virtual" units. Otherwise you can also just write the units in a comment.

With the described situations and the three similar objects named **°C**, it is fully possible to realize the work with each temperature scale: entering the temperature in one scale and converting it, or calculating the input and output values of a

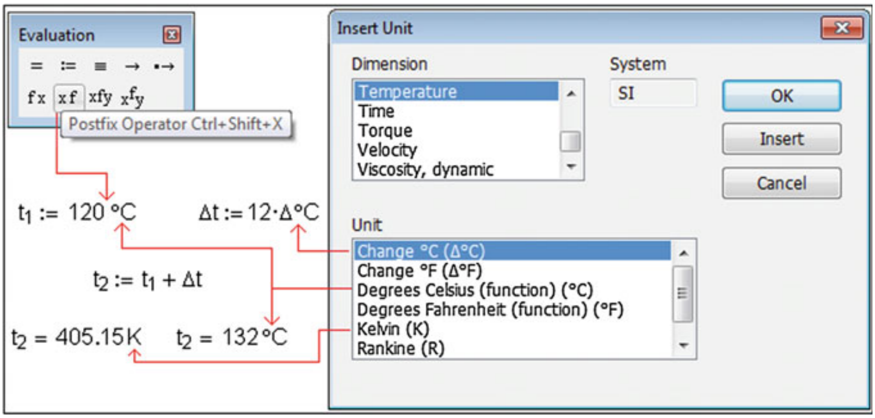

Fig. 2.9 Working with temperature scales in Mathcad 13/14/15

temperature difference. Nevertheless it is necessary to follow a simple but important rule: no matter with which scale we start, the temperature in the formulas should appear only in absolute values (kelvin).

Among Mathcad 13/14/15 and Mathcad Prime is the possibility to work relative temperature scales (Fig. 2.9), but Fig. 2.8 is necessary, since with this method also other relative measurement scales can be introduced to Mathcad. For example, the height of a building measured from the ground level or another point. In addition, it is possible to disable the units of measurement, and leaving only a imitation of units.

Mathcad versions 13/14/15 and also Mathcad Prime have not only standard units of temperature. There is also (temperature difference) **Δ°C** and **Δ°F**. This is done in order to provide the option to use **Δ°C/ΔK** or **Δ°F/Δ°R** equally. It is a tribute to outdated units of measurement. Previously, for example, the thermal conductivity was measured in **W/(m · °C)**, and now in **W/(m · K)**. In Mathcad the unit of thermal conductivity can to be written as: **W/(m · Δ°C)**, but it is better like this: **W/(m · K)**, as **Δ°C = K**.

There is another unit, which could cause some difficulties in Mathcad—it is decibel (dB). Bel is the logarithmic ratio of two one-dimensional physical quantities, and deci means one-tenth. In order to measure a value in decibel, we thus have to build some kind of logarithmic scale of. Usually we measure the strength of sound in decibel. The intensity of sound waves is compared with the lowest sound intensity $\mathbf{I_0}$, which is audible with the human ear (usually $\mathbf{I_0 = 0.01\ W/m^2}$). We can directly calculate the ratio of the measured sound intensity I and the smallest intensity, but the range of values of this ratio so broad that it is necessary to introduce a logarithmic scale with a decimal multiplier—$0.1 \times \log(I/I_0)$. Very loud sounds, e.g., a jack-hammer reaches a level about 100 dB, a conversation in a room is about 60 dB. When using decibels we must select a base from which we use as the base value. In Fig. 2.10 is a scale introduced in relation to the power of the human heart.

In order to work with decibels in Mathcad we introduce two functions named **dB** (the name of one of them is invisible—write with white on white background) and one constant, which is also named **dB**. But these are different objects because they are used by different users (see example above). The operator, which defines the invisible function, "lies" on a colored background in order to make the function name visible. When calling this function with the prefix operator the user cannot see it; instead of seeing **dB p = 2 dB**, he will see **p = 2 dB** (the visible unit **dB** has to be entered manually by the user. Therefore he has to use **dB := 1**. It simulates a unit measurement).

Sometimes when working with decibels is necessary also indicate base (initial value). Thus: **p := 100 dB (re 0.533 W)**, where is just a reference and 0.533 W represents the base. This function requires an infix operator (Fig. 2.11).

Some features can be noted when using dimensionless physical quantities—angles, mass-/ volume-/ and mole-fractions, etc. So, from the 12th version of Mathcad onwards the Steradian become dimensionless (**sr = 1**); before that it was a dimensional quantity **(sr = 1 sr**). The developers on the one hand, restored certainly some logic—radians (the ratio of two lengths) was always dimensionless and Steradian (the ratio of two areas) was dimensional before Mathcad 12 (see above),

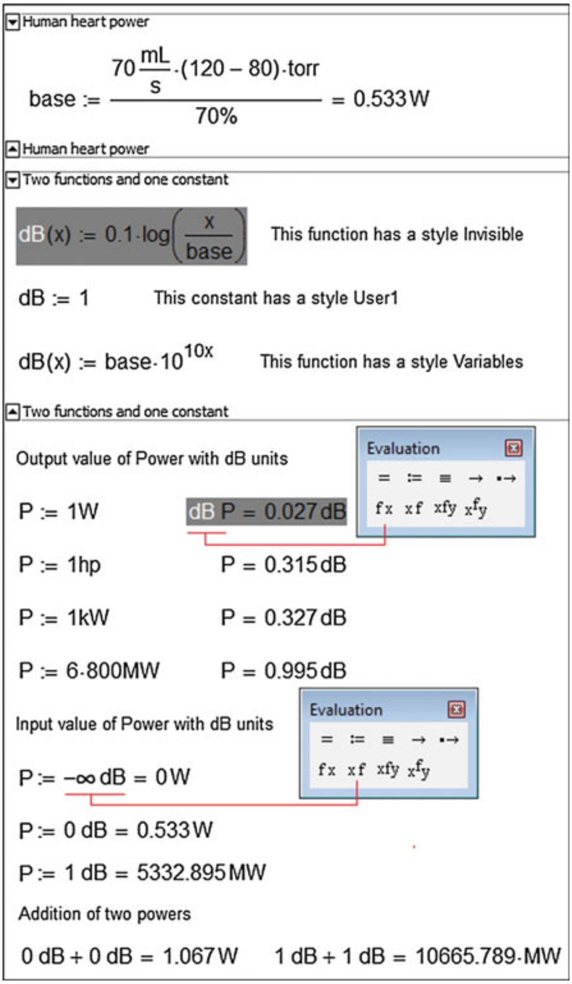

Fig. 2.10 Working with dB

and on the other hand, they opened the possibility of making mistakes such as addition of radian. This problem of lack of control of the dimensions you may encounter when working with other "dimensionless" quantities.

Units referred to the number of pieces (e.g., price per piece) can cause some additional problems. Thus, in the calculation, which is shown in Fig. 2.12, you have to calculate the total surface of 2200 tubes with a diameter of 15 mm. First we have to calculate the surface of one tube. The result we obtain is the surface per piece. This value has afterwards to be multiplied with the total number of pieces to get the final result. The surface per piece is different to the surface of one piece (different units). For example the surface per piece cannot be added to the surface of one piece. To create this new unit of surface per piece we introduce the new unit "piece" (see Fig. 2.12).

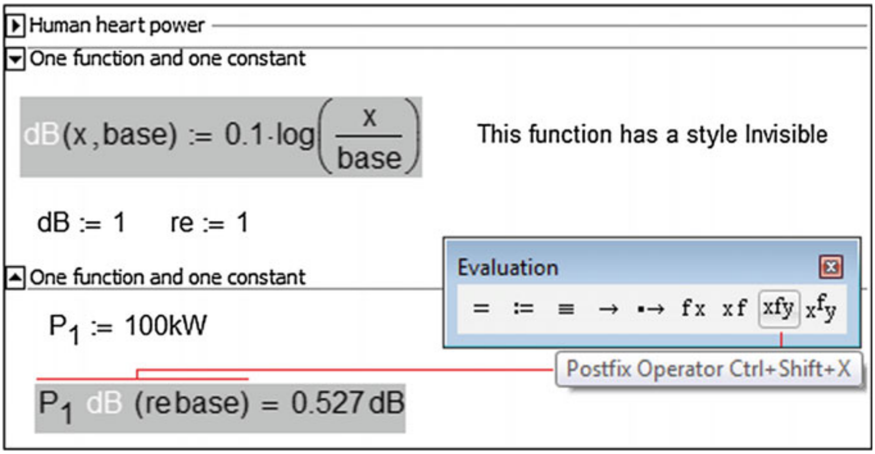

Fig. 2.11 Working with decibels and a reference to the base

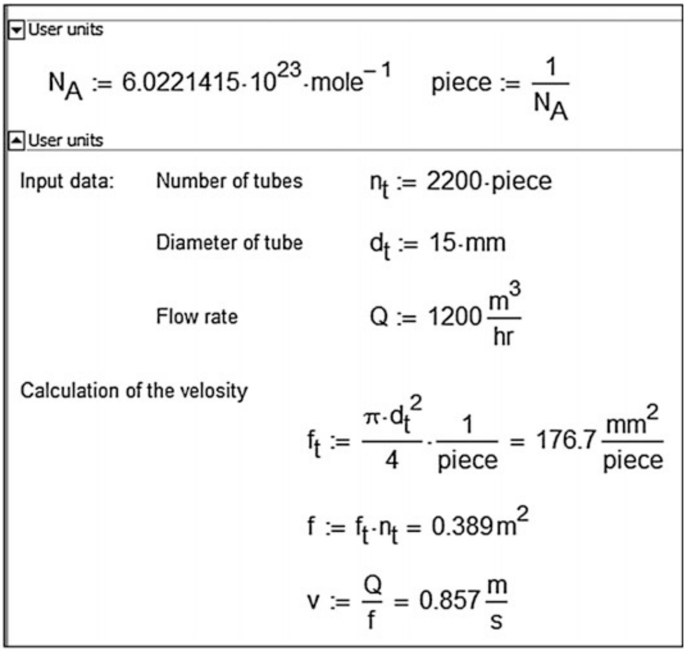

Fig. 2.12 Tube calculation

In some calculations you need two variables with two different units for storing one physical value, for example, the temperature in thermodynamic calculations, where the variable **t** (or **θ**) stores the numerical value in degrees Celsius and **T** in Kelvin. There are tools built into Mathcad in order to eliminate this undue dichotomy.

Without this integrated tool, the calculation of the human heart (Fig. 2.10) would be more complicated and it would be necessary to introduce operators, which translate the units (**mL** and **torr**) in a more popular unit system, for example, in the international SI-system: milliliter (**mL**) → cubic meter (**m^3**), millimeter of mercury (**torr**) → Pascal (**Pa**), etc.

There is another source of possible difficulties when using units. There are, if I may say so, psychological and linguistic reasons related to the unclear definition of physical quantities and units of measurement in everyday speech and even in scientific papers. If two people may have a different understanding of one expression, similar discrepancies can also occur in human-computer "conversations". We say "light year", where the name intends the unit of length—not time, etc.

2.2 Completely Dimensional Functions

We continue with a problem about a pump. Thus, Fig. 2.13 shows an emergency stop mechanism, if the source data contains an error: here was to variable $\mathbf{p}_1$ mistakenly assigned the unit of mass, not pressure.

But in order to avoid the emergency stop of the calculation, it is not only possible to fix the error ($\mathbf{p_1}$ **:= 1 kg**), but also to introduce a new error; ($\mathbf{p_2}$ **:= 20 kg**; see Fig. 2.14).

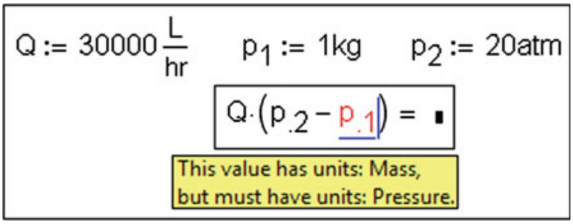

Fig. 2.13 Emergency stop during pump-calculation

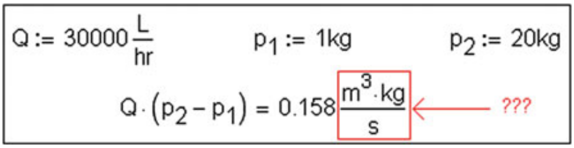

Fig. 2.14 "Fixing" the emergency stop with an error

$$Q := 30000 \qquad p_1 := 1 \qquad p_2 := 20$$

$$Q \cdot (p_2 - p_1) = 5.7 \times 10^5$$

Fig. 2.15 Pump-calculation without units

Of course there will also no emergency stop occur, when using no units at all (see Fig. 2.15).

In principle, all dimensions of the control mechanism should be understood, so that in the formulas the units occur only with certain dimensions—the examples shown in Figs. 2.14 and 2.15 must be completed, so that they stop the calculation, as shown in Fig. 2.13. Another possible problem is the possibility of mixing up the input parameters $\mathbf{p}_1$ and $\mathbf{p}_2$.

The essence of this problem and its possible partial and complete solution, we illustrate in a more common and understandable task—the creation of a function with the name of Vc, which calculates the volume of a cylinder. In Fig. 2.16 shows the function in Mathcad.

The figure shows that if the variables **d** and **h** have the dimensions of length, the function will return the value of volume of the cylinder in m^3. But the same trouble-free function takes also the value of its arguments with any units or even without any units (Fig. 2.17).

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$

$$V_c(20mm, 15mm) = 4.712cm^3$$

Fig. 2.16 Calculating the volume of a cylinder

Fig. 2.17 Function call with different dimensions of the arguments

$$V_c(d,h) := \frac{\pi \cdot d^2}{4} \cdot h$$

$$d := 20 \qquad h := 15kg$$

$$V_c(d,h) = 4712.4kg$$

To find out how to get out of this situation, remember the wonderful principle formulated, which Fourier presented in his classic work "The Analytical Theory of Heat", which was published in 1822. This principle is now called the "principle of dimensional homogeneity" and it states that if the dimensions of each term on both the sides of equation are same, then the physical quantity will be correct. To illustrate the difference between the usual algebraic and "physical" equations, we show an example given in the seminal work of Bridgman ("Regular and Chaotic Dynamics", 2001). There we analyze the problem of a body falling, Bridgman notes that there are at least two equations linking the distance (s), speed (v), time (t) and the acceleration of gravity (g):

$$v = gt$$
$$s = gt^2/2.$$

If we now add both as purely algebraic equation, we obtain the following equation:

$$v + s = gt + gt^2/2$$

That firstly looks plausible, but it is wrong in terms of dimensions—the speed cannot be added to the distance! Having an instrument of controlling the dimensions embedded in Mathcad, we can control the dimension of the entered value by adding a "size zero", namely by adding and subtracting a certain value of a given dimension. Figure 2.18 shows how to do that.

The figure shows that the newly created function, which returns the volume of a cylinder, takes the arguments only in meter (**m**) and returns only values with $\mathbf{m}^3$.

The following example (Fig. 2.19) shows that Mathcad monitors only the dimensions of the variables, but does not check the position of the variables. Therefore it is possible to accidently mix up the variables **d** and **h**.

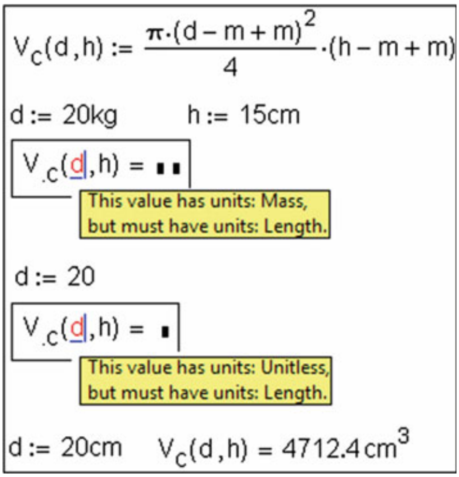

Fig. 2.18 Creating and calling "dimensional" user function

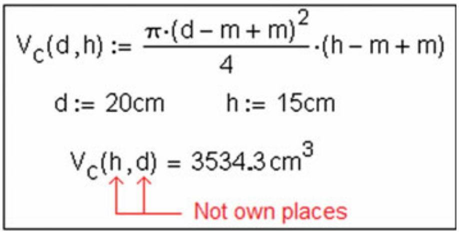

Fig. 2.19 Error while using a user function

The figure shows an error when calling a function that returns the volume of a cylinder, which may occur if the function arguments are reversed: instead of the diameter, we enter the height of the cylinder. This error can be avoided by using further monitoring tools. Here the diameter of the cylinder and its height are two different physical quantities with the same length dimension. First, in the most complete form this idea, how to control the inputs, has been effectively developed in the book Huntley (Huntley G. Dimensional Analysis. Mir, Moscow, 1970), which, in particular, suggested the use of a "vector of dimension".

Thus, the variables have to be connected to the dimensions (length–meters, feet, miles, etc., weight–pounds, pounds, etc.), and to the physical quantities. It should be assumed that the two variables (diameter of the cylinder and its height) have two different units of measurement, and their "difference", in particular, should be reflected in the fact that these parameters cannot, for example, be added.

In Mathcad are eight dimensions integrated: length, time, mass, current, temperature, light intensity, the amount of substance, and currencies (only in the 14th and 15th versions and Mathcad Prime). In order to avoid the possibility of mixing up the input parameters we introduce two different lengths. The diameter is connected to the unit **m-d** (m = meter, d = diameter) and the height is connected to the unit **m-h** (m = meter, h = height). Basically the new unit **m-h** has not the unit meter. Therefore we divide in the function $\mathbf{V}_c$ the input parameter by his unit **m-h**. Afterwards the function is multiplied with $\mathbf{m}^3$ in order to obtain the correct result with correct units (Fig. 2.20).

When you now swap wittingly or unwittingly the arguments of the function the calculation is interrupted by an error message, which can be described as follows: "The inputs are swapped".

There are also real thermodynamic problems, which can be solved with Mathcad, whereby we need or, at least, it would be desirable to handle different physical variables, which have the same dimension.

You need to calculate the thermal efficiency of a combined-cycle (see Chap. 14). To solve this problem, you need functions, which return the thermodynamic properties of the working fluids. These functions can be integrated to Mathcad by using the already mentioned WaterSteamPro software (www.wsp.ru)—see Fig. 2.21.

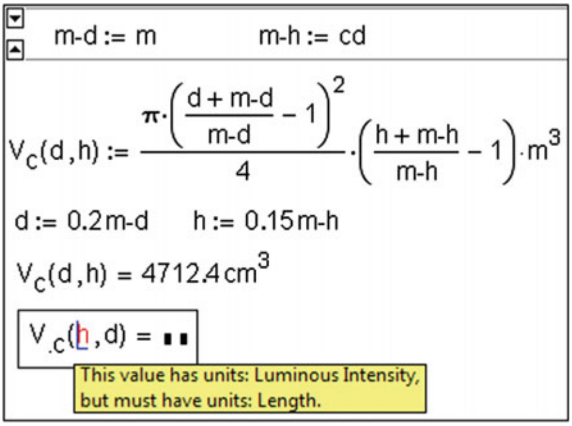

Fig. 2.20 Calculation of the volume of a cylinder with full control

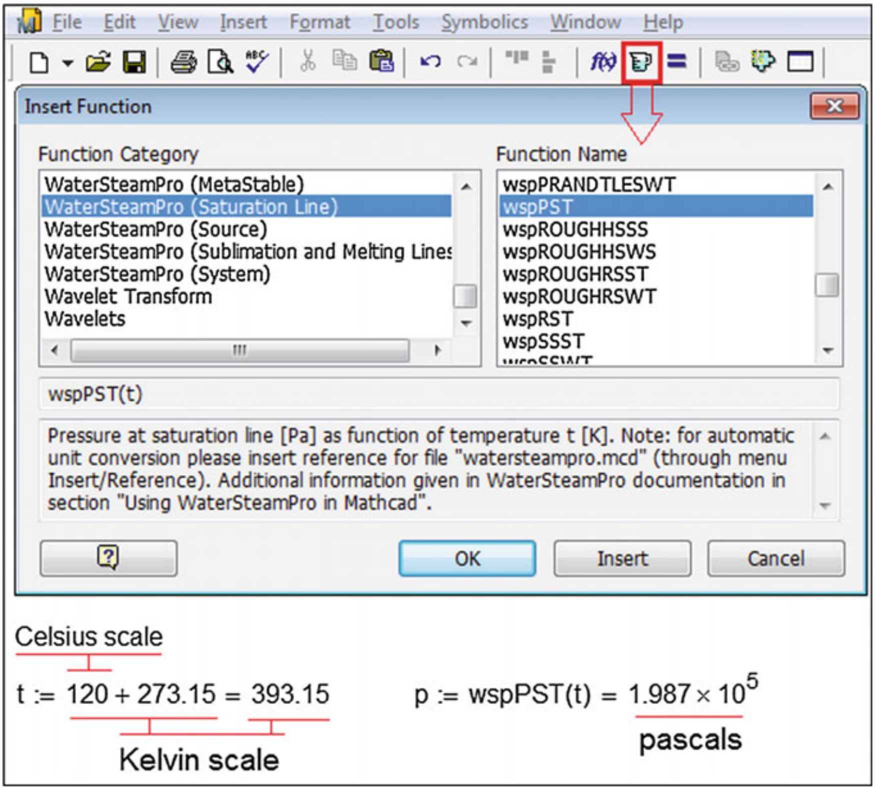

Fig. 2.21 Using WaterSteamPro in Mathcad

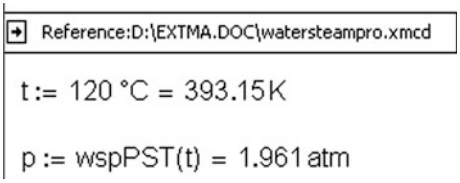

Fig. 2.22 Using WaterSteamPro with units

Figure 2.21 shows how you can calculate the boiling point of water for a given pressure by using one of the functions from WaterSteamPro—function wspTSP. Moreover, all values are non-dimensional, but it is meant that the pressure is expressed in pascal, and the temperature is measured in kelvin. All functions from WaterSteamPro by default (without referencing to "watersteampro.xmcd") use dimensionless arguments but expressed in SI units or their mixture.

But Mathcad software supports units, and this mechanism is necessary to solve our problem and indeed in many other heat and power calculations.

In order to be also able to use units for WaterSteamPro it is necessary to make a reference link to **watersteampro.xmcd**, included in the WaterSteamPro (Fig. 2.22).

What contains this file? There are firstly some additional units definitions, as well as various constants needed for calculations in heat and power engineering (the universal gas constant, critical points of water and steam, etc.). Secondly, and most importantly, it makes units available for WaterSteamPro.

In the programs Mathcad Prime 1 and 2, WaterSteamPro cannot be integrated through the mechanism of DLL. Such an opportunity came only in the third version of Mathcad Prime. Figure 2.23 shows how to call a function named **wspPST**, which returns the saturation pressure of water and steam as a function of temperature. Calling this function in Mathcad 15 is shown in Fig. 2.21 ("dimensionless" call) and Fig. 2.22 ("dimensional" call through the supporting document watersteamepro. mcd). The mechanism of function redefining shown on Fig. 2.24 on the example made im Mathcad 15 won't work in Mathcad Prime 3 environment. WaterSteamPro functions, that ypu can see on Fig. 2.23 must be "dimensionless" or supplemented by the corresponding basic units (such as **Pa** for pressure). We will examine this methods more closely in the Chap. 9 on the example of nuclear power plant calculation.

However in Mathcad 15 it isn't necessary to insert reference on watersteampro. xmcd (which redefines all the functions of WaterSteamPro program), instead, it is possible to manually redefine only those functions that are used in the users' project taking into account, that different physical values might have same units. For example in Chap. 14 (calculation of the thermal efficiency of the CCGT unit) reader will encounter with different physical values—mass of the gas and mass of the water/steam, which measures in one and the same units of mass.

This is one of the key aspects of working with units.

In thermophysical calculations reader may face values that have identical unit, but different physical sense. If this fact doesn't taken into account, it may lead to the errors during calculations in the Mathcad environment, which we mentioned before.

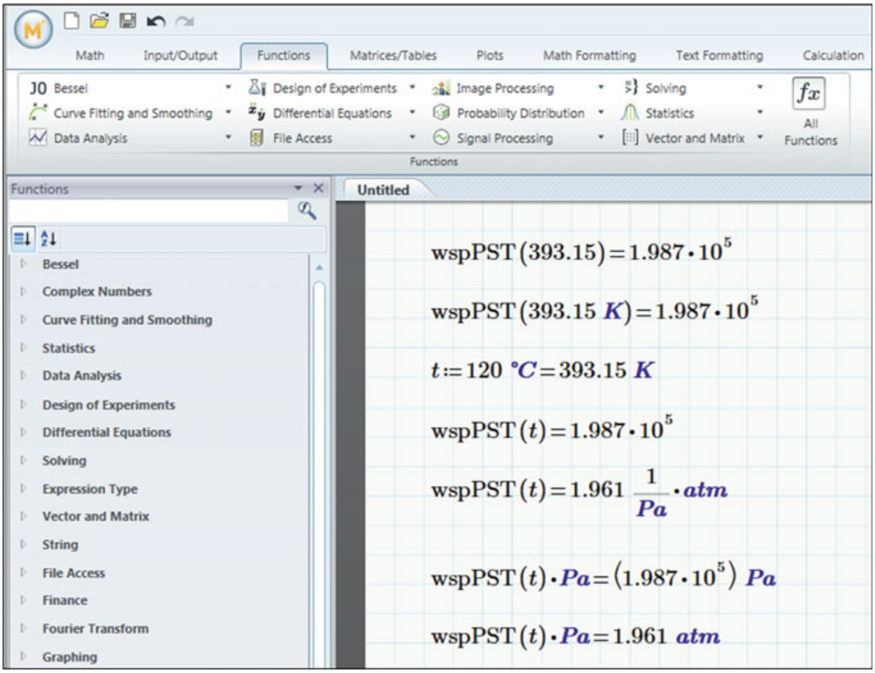

Fig. 2.23 WaterSteamPro functions with units in Mathcad Prime 3

Example In the binary thermodynamic cycle (CCGT unit for example) among the other values appear values such as specific enthalpy of the first working medium and specific enthalpy of the second working medium. Both of these values are typically measured in kJ/kg, but they have different kilograms. To avoid mistakes of mixing different physical values in such calculations, it is recommended to use built-in kilograms for one of the values and for the second assign basic SI unit, that isn't used in users' calculation (for example candela—**cd**). In that case value **m**, that usually records ratio of the first working body flow rate to the flow rate of the second in binary thermodynamic cycles, will be dimensional value, and it will help to avoid some typical mistakes.

The technology of entering different physical values with same units into calculation shown on Fig. 2.24.

As you can see at the top of Fig. 2.24 introduced two units of mass: for the measurement of water vapor and gas—the two working fluids of the combined-cycle plant. Kilograms of water/steam equal to the built-in Mathcad unit of mass (**kg**). Kilograms of gas equal a specific unit, which in this calculation is not used—the unit of luminous intensity **cd** (candela).

The next stage in the calculation in Fig. 2.24 is the input of initial data. Notice that in values of mass flow rate are used kilograms of water vapour and kilograms of gas for different physical quantities that have the same mass dimension.

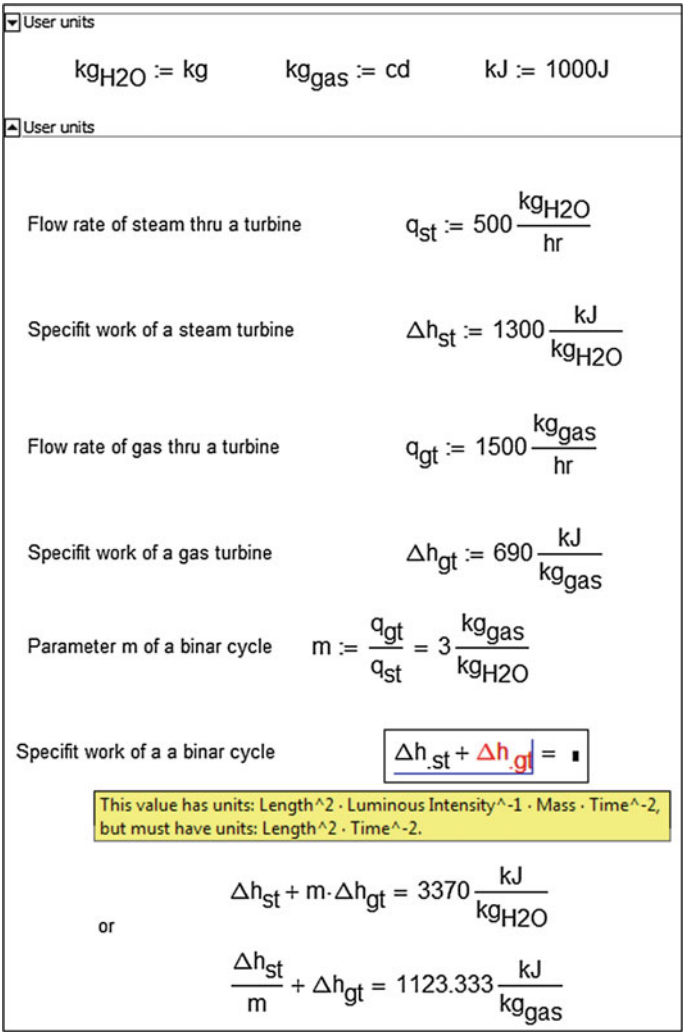

Fig. 2.24 Performance assessment of a combined cycle plant

This allows user to have in his calculation the value of **m** (the ratio of the flow rate of one of the working medium to the flow rate of another working body) not dimensionless, but dimensional. This helps to avoid some errors in the calculation of binary power cycles, one of which is shown in Fig. 2.24: if we had not introduced two units of mass for the two working fluids, the operator $\mathbf{\Delta h_{st} + \Delta h_{gt}}$ would not have been interrupted by the error message, and would have given an incorrect result. The correct result (specific work of the combined cycle) is calculated from the last two operators shown in Fig. 2.24.

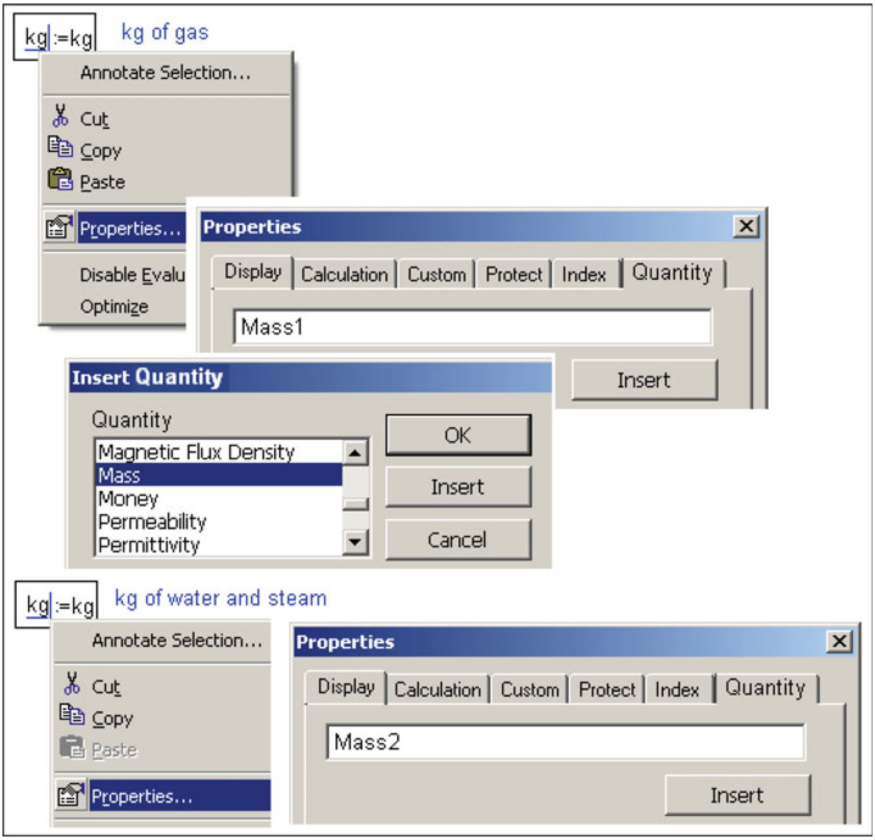

Fig. 2.25 Demonstration of the toolkit for creating different quantities with the same unit

One solution of Mathcad problems with different physical quantities of the same dimension can be offered through an additional tab "Quantity" in the dialog box "Properties", which occurs if we press the right mouse button on the name of the newly created function or variable. In the text box of this tab can be written the desired dimension/index—the base (weight, volume, etc.) or component (air, gas, etc.). Thus the name can be supplemented by the numbers 1, 2, 3, etc., dividing one from the other physical value. This technique is displayed in Fig. 2.25.

In the calculation shown in Fig. 2.24 dimensionless variable m (the mass flow in a gas turbine divided by the mass flow of steam in a steam turbine) is transformed into a ("pseudo"-) dimensional variable. And this is, as we have seen, good: the less dimensionless quantities, the more monitoring tools can be used. But in terms of dimensionless quantities we also have other ones—the degree of dryness of steam (dry weight of steam referred wet steam) and the three values of thermal efficiency (**η**).

If the numerators and denominators in these fractions have the same dimension, these ratios must be dimensionless. If the numerators and denominators of these fractions are regarded to different physical quantities, these values (**m**, **η**, etc.) should be "pseudo"-dimensional. It is possible to use, for example, the technique shown in Fig. 2.25.

http://www.springer.com/978-3-319-26673-2

Thermal Engineering Studies with Excel, Mathcad and Internet
Ochkov, V.; Orlov, K.; Voloshchuk, V. - Rogalev, N. (Ed.)
2016, XV, 307 p. 284 illus., 221 illus. in color.,
Hardcover
ISBN: 978-3-319-26673-2